

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-242676

(43) 公開日 平成11年(1999) 9月7日

(51) Int.Cl.⁶

G 0 6 F 17/30

識別記号

F I

G 0 6 F 15/40

15/403

3 7 0 A

3 4 0

3 1 0 A

審査請求 未請求 請求項の数20 O L (全 89 頁)

(21) 出願番号

特願平10-43187

(22) 出願日

平成10年(1998) 2月25日

(71) 出願人

000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者

岡本 卓哉

神奈川県川崎市幸区鹿島田890番地 株式

会社日立製作所情報通信開発本部内

(72) 発明者

高橋 亨

神奈川県川崎市幸区鹿島田890番地 株式

会社日立製作所情報通信開発本部内

(72) 発明者

川口 久光

東京都千代田区神田駿河台四丁目6番地

株式会社日立製作所内

(74) 代理人

弁理士 小川 勝男

最終頁に続く

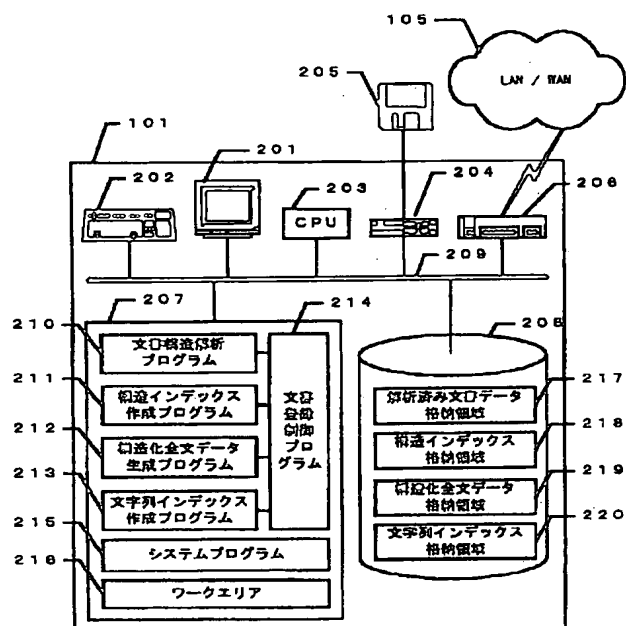
(54) 【発明の名称】 構造化文書登録方法、検索方法、およびそれに用いられる可搬型媒体

(57) 【要約】

【課題】 構造化文書を対象とした構造指定検索において、論理要素の文書中における出現位置に関する条件を指定できるようにして精度の高い構造指定検索を可能にすること。

【解決手段】 文書をデータベースに登録する際、各登録対象文書の持つ論理構造を重ね合わせ、文書中での出現位置が等しい構造要素を単一のメタノードによって代表させた構造インデックスを作成し、文書検索時には構造インデックスを参照して指定された構造条件を満足するメタノードの集合を求め、それらのメタノードの識別子をキーとして文字列インデックスを検索することにより、指定条件を満たす文書群を求める。これにより、構造化文書の集合からなる文書データベース上において、精度の高い構造指定検索が可能となる。

図2



1

【特許請求の範囲】

【請求項1】 予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理が、

登録対象文書の持つ論理構造を解析して得られる解析済み文書データを文書データベースに登録する解析済み文書データ生成登録ステップと、

各登録対象文書の持つ論理構造を順次重ね合わせ、文書中における出現位置および種別が同一である構造要素群を単一のメタノードによって代表させた構造インデックスを作成し、該構造インデックスの木構造を構成する各メタノードに、該メタノードを一意に識別する文脈識別子を与える構造インデックス作成ステップを有することを特徴とした構造化文書の登録方法。

【請求項2】 予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理が、

各登録対象文書について、該文書中に含まれる各文字列データから、所定の部分文字列と、該部分文字列の登録対象文書中における文字位置情報と、前記登録対象文書を文書データベース中で一意に識別する文書識別子と、前記部分文字列を含む文字列データを前記構造インデックス中で代表するメタノードの文脈識別子とを抽出し、前記文字位置情報および前記文書識別子ならびに前記文脈識別子からなる構造化文字位置情報を生成し、前記部分文字列と前記構造化文字位置情報との対応関係を登録して文字列インデックスを更新する文字列インデックス更新ステップを有することを特徴とした構造化文書の登録方法。

【請求項3】 予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理の過程で、

各登録対象文書の持つ論理構造を順次重ね合わせ、文書中における出現位置および種別が同一である構造要素群を単一のメタノードによって代表させた構造インデックスを作成し、該構造インデックスの木構造を構成する各メタノードに、該メタノードを一意に識別する文脈識別子を与える構造インデックス作成ステップにおいて、二つの登録対象文書の持つ木構造を重ね合わせる際に、両文書の木構造を構成する二つのノードを比較したときに、両ノードが共に木構造の根であれば両ノードは互いに対応しているとは見なし、

両ノードが根でない場合には、両ノードの上位ノードが互いに対応し、かつ両ノードの種別が一致し、かつ種別の等しい兄弟ノードの並びの先頭から正順に数えた出現順序が等しい場合に両ノードは対応しているとは見なし、対応している両ノードを単一のメタノードによって代表

2

するように前記構造インデックスを構築することを特徴とした構造化文書の登録方法。

【請求項4】 予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理の過程で、

各登録対象文書の持つ論理構造を順次重ね合わせ、文書中における出現位置および種別が同一である構造要素群を単一のメタノードによって代表させた構造インデックスを作成し、該構造インデックスの木構造を構成する各メタノードに、該メタノードを一意に識別する文脈識別子を与える構造インデックス作成ステップにおいて、二つの登録対象文書の持つ木構造を重ね合わせる際に、両文書の木構造を構成する二つのノードを比較したときに、両ノードが共に木構造の根であれば両ノードは互いに対応しているとは見なし、

両ノードが根でない場合には、両ノードの上位ノードが互いに対応し、かつ両ノードの種別が一致し、かつ種別の等しい兄弟ノードの並びの末尾から逆順に数えた出現順序が等しい場合に両ノードは対応しているとは見なし、対応している両ノードを単一のメタノードによって代表するように前記構造インデックスを構築することを特徴とした構造化文書の登録方法。

【請求項5】 予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理の過程で、

各登録対象文書の持つ論理構造を順次重ね合わせ、文書中における出現位置および種別が同一である構造要素群を単一のメタノードによって代表させた構造インデックスを作成し、該構造インデックスの木構造を構成する各メタノードに、該メタノードを一意に識別する文脈識別子を与える構造インデックス作成ステップにおいて、二つの登録対象文書の持つ木構造を重ね合わせる際に、両文書の木構造を構成する二つのノードを比較したときに、両ノードが共に木構造の根であれば両ノードは互いに対応しているとは見なし、

両ノードが根でない場合には、両ノードの上位ノードが互いに対応し、かつ両ノードの種別が一致し、かつ種別の等しい兄弟ノードの並び中において、両ノードが共に該並びの先頭位置にあるか、または両ノードが共に先頭以外の位置にある場合に、両ノードは対応しているとは見なし、

対応している両ノードを単一のメタノードによって代表するように前記構造インデックスを構築することを特徴とした構造化文書の登録方法。

【請求項6】 予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理の過程で、

10

20

30

40

50

3

各登録対象文書の持つ論理構造を順次重ね合わせ、文書中における出現位置および種別が同一である構造要素群を単一のメタノードによって代表させた構造インデックスを作成し、該構造インデックスの木構造を構成する各メタノードに、該メタノードを一意に識別する文脈識別子を与える構造インデックス作成ステップにおいて、二つの登録対象文書の持つ木構造を重ね合わせる際に、両文書の木構造を構成する二つのノードを比較したときに、両ノードが共に木構造の根であれば両ノードは互いに対応しているとし、
両ノードが根でない場合には、両ノードの上位ノードが互いに対応し、かつ両ノードの種別が一致し、かつ種別の等しい兄弟ノードの並び中において、両ノードが共に該並びの末尾位置にあるか、または両ノードが共に末尾以外の位置にある場合に、両ノードは対応しているとし、
対応している両ノードを単一のメタノードによって代表するように前記構造インデックスを構築することを特徴とした構造化文書の登録方法。

【請求項 7】 予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、
文書の登録を行う処理の過程で、
登録対象文書の持つ論理構造を解析して得られる解析済み文書データを文書データベースに登録する際に、
該解析済み文書データ中から、検索対象としては不適切な構造および内容文字列を抽出し、該構造および内容文字列を解析済み文書データ中から削除した後に、
該解析済み文書データを文書データベースに登録するステップを有することを特徴とした構造化文書の登録方法。

【請求項 8】 請求項 1 に記載の構造化文書の登録方法において、
前記解析済み文書データ生成登録ステップおよび前記構造インデックス作成ステップを実行することにより作成された構造インデックスを記録した可搬型媒体。

【請求項 9】 請求項 1 に記載の構造化文書の登録方法において、
前記解析済み文書データ生成登録ステップおよび前記構造インデックス作成ステップを実行する機能を備えたコンピュータプログラムを記録した可搬型媒体。

【請求項 10】 請求項 2 に記載の構造化文書の登録方法において、
前記文字列インデックス更新ステップを実行することにより作成された前記文字列インデックスを記録した可搬型媒体。

【請求項 11】 請求項 2 に記載の構造化文書の登録方法において、
前記文字列インデックス更新ステップを実行する機能を備えたコンピュータプログラムを記録した可搬型媒体。

4

【請求項 12】 予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の検索方法であって、
文書の検索を行う処理が、
構造インデックスを参照し、指定された構造条件を満たす文脈識別子の集合を決定する構造条件判定ステップと、
検索タームから所定の部分文字列を抽出し、文字列インデックスを参照して該部分文字列に対応する構造化文字位置情報の集合を抽出する構造化文字位置情報抽出ステップと、

前記構造化文字位置情報の集合中から、前記構造条件判定ステップで決定した集合中に含まれる文脈識別子を持ち、かつ前記検索ターム上における部分文字列の並びと同じ位置関係をもつ構造化文字位置情報を抽出するインデックス検索ステップを有することを特徴とした構造化文書の検索方法。

【請求項 13】 請求項 12 に記載の構造化文書の検索方法において、
前記構造条件判定ステップおよび前記構造化文字位置情報抽出ステップおよび前記インデックス検索ステップを実行する機能を備えたコンピュータプログラムを記録した可搬型媒体。

【請求項 14】 予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、
文書の登録を行う処理が、
登録される文書の持つ論理構造を解析して得られる解析済み文書を作成するステップと、

文書の要素型名と構造の種別の対応を定義した種別定義テーブルの内容を解析して、同じ種別とみなされる構造を得るステップと、

最上位の構造が同じ種別であるとみなされる、複数の解析済み文書の論理構造を重ね合わせ、2つ以上の文書において同じ位置、種別とみなされる構造を1つのメタノードに代表させ、1つの文書のみに出現するとみなされる構造は、その構造を表わす1つのメタノードを生成することで、最上位の構造が同じとみなされる複数の登録文書ごとに、共通の構造情報をメタノードを要素とする木構造で表わす構造インデックスを作成するステップと、
最上位の構造の種別が同じとみなされる複数の登録文書ごとに作成された複数の構造インデックスの最上位構造を表わすメタノードの共通の上位構造として、登録文書に対応する構造を持たないルートメタノードを作成し、各登録文書の最上位構造を表わすメタノードは、ルートメタノードの子構造とした、ルートメタノードを最上位ノードとするメタ構造インデックスを作成するステップと、
ルートメタノードを含めてメタ構造インデックスを構成するメタノードを一意に識別する文脈識別子を各構造インデックスを構成するメタノードに与えるステップと、

5

各メタノードに対応した各登録文書の構造内の文字列を登録文書の識別情報と文脈識別子の情報と共に文字列インデックスに登録するステップを有することを特徴とした構造化文書の登録方法

【請求項 1 5】 予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理が、

登録される文書の持つ論理構造を解析して得られる解析済み文書を作成するステップと、

文書の要素型名と構造の種別の対応を定義した種別定義テーブルの内容を解析して、同じ種別とみなされる構造を得るステップと、

最上位の構造が同じ種別であるとみなされる、複数の解析済み文書の論理構造を重ね合わせ、2つ以上の文書において同じ位置、種別とみなされる構造を1つのメタノードに代表させ、1つの文書のみ出現するとみなされる構造は、その構造を表わす1つのメタノードを生成することで、最上位の構造が同じとみなされる複数の登録文書ごとに、共通の構造情報をメタノードを要素とする木構造で表わす構造インデックスを作成するステップと、最上位の構造の種別が同じとみなされる複数の登録文書ごとに作成された複数の構造インデックスの最上位構造を表わすメタノードの共通の上位構造として、登録文書に対応する構造を持たないルートメタノードを作成し、各登録文書の最上位構造を表わすメタノードは、ルートメタノードの子構造とした、ルートメタノードを最上位ノードとするメタ構造インデックスを作成するステップと、ルートメタノードを含めてメタ構造インデックスを構成するメタノードを一意に識別する文脈識別子を各構造インデックスを構成するメタノードに与えるステップと、各メタノードに対応した各登録文書の構造内の文字列を登録文書の識別情報と文脈識別子の情報と共に文字列インデックスに登録するステップと、

同じ種別を持つ、異なる位置に出現する構造に対する共通名称を別名として設定し、該別名とメタノードの文脈識別子を対応付けた、別名構造インデックスを生成するステップを有することを特徴とした構造化文書の登録方法

【請求項 1 6】 予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理が、

登録される文書の持つ論理構造を解析して得られる解析済み文書を作成するステップと、

文書の要素型名と構造の種別の対応を定義した種別定義テーブルの内容を解析して、同じ種別とみなされる構造を得るステップと、

最上位の構造が同じ種別であるとみなされる、複数の解析済み文書の論理構造を重ね合わせ、2つ以上の文書において同じ位置、種別とみなされる構造を1つのメタノ

6

ードに代表させ、1つの文書のみ出現するとみなされる構造は、その構造を表わす1つのメタノードを生成することで、最上位の構造が同じとみなされる複数の登録文書ごとに、共通の構造情報をメタノードを要素とする木構造で表わす構造インデックスを作成するステップと、各構造インデックスを構成するメタノードを一意に識別する文脈識別子を構造インデックスを構成するメタノードに与えるステップと、

10 各構造インデックスを識別するための構造インデックス識別子を構造インデックスに与えるステップと、

各メタノードに対応した各登録文書の構造内の文字列を登録文書の識別情報と構造インデックス識別子と文脈識別子の情報と共に文字列インデックスに登録するステップと、

20 複数の構造インデックスにまたがって同じ種別であるとみなされる特定の種別を持つ構造に対する共通名称を別名として設定した別名テーブル共通名称定義テーブルを読み出し、共通名称と各構造インデックスの構造インデックス識別子およびメタノードの文脈識別子を対応付けた、共通名称構造インデックスを生成するステップ同じ種別を持ち異なる構造インデックス、または位置に出現する構造に対する共通名称を別名として設定し、該別名と構造インデックス識別子およびメタノードの文脈識別子を対応付けた、別名構造インデックスを生成するステップを有することを特徴とした構造化文書の登録方法

【請求項 1 7】 予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理が、

30 登録される文書の持つ論理構造を解析して得られる解析済み文書を作成するステップと、

文書の要素型名と構造の種別の対応を定義した種別定義テーブルの内容を解析して、同じ種別とみなされる構造を得るステップと、

各文書の最上位構造の上に全文書に共通な仮の最上位構造を追加した文書に変換するステップと、

該仮の最上位構造を最上位の構造として、複数の解析済み文書の論理構造を重ね合わせ、2つ以上の文書において同じ位置、種別とみなされる構造を1つのメタノードに代表させ、1つの文書のみ出現するとみなされる構造は、その構造を表わす1つのメタノードを生成することで、共通の構造情報をメタノードを要素とする木構造で表わす構造インデックスを作成するステップと、

構造インデックスを構成するメタノードを一意に識別する文脈識別子をメタノードに与えるステップと、

各メタノードに対応した各登録文書の構造内の文字列を登録文書の識別情報と文脈識別子の情報と共に文字列インデックスに登録するステップを有することを特徴とした構造化文書の登録方法

【請求項 1 8】 請求項 1 2 に記載の構造化文書の検索方法であって、

50

7

文書の検索を行う処理が、
構造インデックスを参照し、指定された構造条件を満たす文脈識別子の集合を決定する構造条件判定ステップにおいて、

文書の要素型名と構造の種別の対応を定義した種別定義テーブルを用いて、検索条件に記載された要素型名を種別情報に変換した上で、構造インデックスを参照して、種別情報に適合する文脈識別子の集合を決定することを特徴とした構造化文書の検索方法。

【請求項 1 9】請求項 1 2 に記載の構造化文書の検索方法であって、

文書の検索を行う処理が、
構造インデックスを参照し、指定された構造条件を満たす文脈識別子の集合を決定する構造条件判定ステップにおいて、

共通名称インデックスを用いて検索条件に記載された要素型名を基に、構造インデックスの対応するノードの位置を取得することにより、種別情報に適合する文脈識別子の集合を決定することを特徴とした構造化文書の検索方法。

【請求項 2 0】請求項 1 に記載の構造化文書の検索方法であって、

文書の登録を行う処理において、登録文書の構造に付けられた名称を構造インデックスにおける構造の重ね合わせのために種別に変換するステップを有することを特徴とした構造化文書の登録方法

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】本発明は、コンピュータ装置を用いた文書検索システムや文書管理システム等における文書登録方法および文書検索方法に係わり、特に、各文書がそれぞれ論理的な構造を備えている構造化文書の集合を対象として、特定の文書内容の検索を高速に行うことができるようにした構造化文書の登録方法、検索方法、およびそれに用いられる可搬型媒体に関する。

【0 0 0 2】

【従来の技術】情報化社会の本格的な進展に伴い、ワードプロセッサ、パーソナルコンピュータ等を用いて作成された電子化文書情報が爆発的な勢いで増加しつつある。このような状況下において、蓄積された膨大な電子化文書群の中から、必要とする情報を含んだ文書を高速かつ確実に検索したいという要求が高まっている。

【0 0 0 3】このような要求に応える技術として全文検索がある。全文検索では登録対象文書中のテキスト全体を計算機システムに投入してデータベース化し、該データベース上で指定文字列（以後、検索タームと呼ぶ）を直接検索するので、キーワードを必要とせず、原理的に検出漏れのない検索が可能となる。

【0 0 0 4】また、例えば S G M L (ISO 8879:1986, Standard Generalized Markup Language) で記述さ

8

れた文書など、文書を構成する個々の論理的な構造要素が識別できる文書（以下、構造化文書と呼ぶ）を対象として、論理構造に関する条件を検索条件中に付加した検索（以下、構造指定検索と呼ぶ）を行うことにより、精度の高い検索を実現することができる。

【0 0 0 5】構造指定検索を可能にする方式としては、例えば特開平 8 - 1 4 7 3 1 1 号公報に示す発明（以下、公知例 1 と呼ぶ）で提案した検索方法がある。以下、公知例 1 の概要について説明する。

10 【0 0 0 6】公知例 1 の構造化文書検索方法は、文書を登録する際、まず登録文書をそのまま本文として検索用データベースに登録する。

【0 0 0 7】次に、登録した本文について各論理構造の先頭を表わす特定の文字列（これを公知例 1 では前方マーカと呼ぶ）および末尾を表わす特定の文字列（これを公知例 1 では後方マーカと呼ぶ）を検出することにより論理構造の識別処理を行うとともに、本文を論理構造ごとに分割する処理を行う。例えば、電子出願特許明細書の場合には、論理構造「要約」の範囲を示す前方マーカとして“<SDO ABJ>”を、後方マーカとして“</SDO>”を検出し、これらによって囲まれるテキストを「要約」に対応する本文として切り出す。他の論理構造についても同様な切り出しを行い、本文を論理構造ごとに分割する。

【0 0 0 8】次に、分割された各論理構造に対応する本文について、それぞれ凝縮本文の作成処理を行う。すなわち、「要約」については「要約」に関する本文テキストを単語単位に部分文字列に分割し、分割した部分文字列間で相互に包含関係を調べ、他の部分文字列に含まれる文字列を排除することにより「要約」に関する凝縮本文を作成する。他の論理構造についても同様な処理を行うことにより論理構造別の凝縮本文を作成し、これを凝縮本文ファイルとして検索用データベースに登録する。

【0 0 0 9】次に、本文中に現れた文字の文字コードに対応するビットに「1」を設定することにより文字成分表を作成し、これを検索用データベースに文字成分表ファイルとして登録する。

【0 0 1 0】このようにして検索用データベースを構築した後、公知例 1 では次のようにして文書検索処理を行う。

【0 0 1 1】はじめに、指定された検索タームを文字単位に分解し、検索タームを構成するすべての文字がともに含まれる文書を上記文字成分表を参照して抽出する。

【0 0 1 2】次に、各論理構造に対応する凝縮本文ファイルのうち、検索対象として指定された論理構造に関する凝縮本文ファイルをサーチ対象として選択するとともに、その中で文字成分表サーチによって抽出された文書の凝縮本文だけを対象としてサーチすることにより、指定された論理構造中に指定された検索タームが含まれている文書を抽出する。指定された検索条件式に複数の検

50

索ターム間の本文中での位置関係が指定されていない場合にはここで検索処理を終了する。他方、そのような位置関係が指定されている場合には凝縮本文サーチの結果抽出された文書に対応する本文の内容を読み、指定された検索タームがすべて含まれ、かつ検索ターム間の位置関係について、指定された条件が満たされるもののみを抽出する。

【0013】このようにして、公知例1の方法によれば、大規模なテキストデータベースに対する実用的な検索速度を保ち、かつ構造指定検索を可能にすることができる。

【0014】

【発明が解決しようとする課題】前記公知例1に示す従来技術によって、一定の構造指定検索が可能となる。しかし、公知例1による構造指定検索では意図した構造条件に従った検索ができない場合がある。

【0015】公知例1の方式では、登録対象文書の持つ構造を予め定めておいたいくつかの部分構造に切り分け、各部分構造ごとに凝縮本文ファイルを作成し、検索時には、部分構造の構造名と凝縮本文ファイルとの対応を定義したテーブルを参照して検索対象とすべき凝縮本文ファイルの集合を決定し、該集合中に含まれる凝縮本文ファイルだけを対象として検索を行うことにより構造指定検索を実現する。

【0016】この方式では、テキストデータベースを構築する段階で、将来どのような構造条件を指定した検索が行われるかを想定し、そのような条件に対応した検索が可能となるように凝縮本文ファイルの切り分けを行うので、データベース構築時に想定していなかった構造条件を指定した検索は不可能となる。

【0017】例えば、文書が“要約”と“本体”の二つの論理要素（以下、要素と呼ぶ）から構成され、“本体”はさらに任意個数の“節”の繰り返しからなり、各“節”は1個の“節題”と任意個数の“段落”からなる場合を考える。このような構造を持つ文書群からなるテキストデータベースを構築する際に、凝縮本文ファイルを“要約”に対応するものと“本体”に対応するものの二つに分けて作成したとすると、「節題中に文字列〇〇を含む文書群を求める」という条件を満たす構造指定検索は不可能となってしまう。

【0018】もちろん、“本体”全体で1個の凝縮本文ファイルとする代わりに、これをさらに“節題”と“段落”とに分けて凝縮本文ファイルを作成しておけばこの条件に対応することはできる。しかし、仮にこのようにファイルを構成しても、例えば「最初の節の内部（節題または段落中）に文字列〇〇を含む文書群を求める」、あるいは「節中で最後の段落に文字列××を含む文書群を求める」というような構造条件には対応できない。このような順位指定付きの構造条件に対応しようとする

と、予め節および段落の出現順位ごとに別々に凝縮本文

ファイルを用意しておくことが必要となるが、節および節内の段落はそれぞれ任意個数出現しうるため凝縮本文ファイルの数が極めて多数となるだけでなく、公知例1は出現順位に関する任意の指定を含んだ構造条件から細分化された凝縮本文ファイルの集合への対応付けを行う手段を備えていないので、実際にはこのような条件を満たす検索は不可能となる。

【0019】このように、前記従来技術では、文書中に現れた論理要素の出現位置に関する条件を構造条件指定中に含めることができないため、精度の高い構造指定検索ができないという問題があった。

【0020】本発明の目的は、前記従来技術の持つ上記問題点を解決し、高精度でかつ効率のよい構造指定検索を行う機能を提供することにある。

【0021】さらに、前記従来技術では、あらかじめ定められた1つの構造を持った文書群に対する構造指定検索しか実現できない。

【0022】つまり、SGMLなどの構造化文書は、文書型定義（DTD: Document Type Definition）によりあらかじめ定められた構造を持つ文書である。したがって、特定の文書型定義に従った文書群に対する構造指定検索を行う場合は、出現しうる構造指定条件すべてに対応するように、文書を構造毎に切り分けることで、構造指定検索が可能となる。

【0023】しかし、文書型定義は1つだけではない。例えば、論文、報告書などはそれぞれ異なる文書型定義を持っている。このように、構造化文書では、その文書の目的に応じて様々な文書構造を持っており、その文書構造に合わせた文書型定義が作成される。

【0024】これらの文書を文書型定義毎にグループ化して登録することで、個々のグループにおいては、構造指定検索が可能となるが、全グループに共通して出現しうる構造を指定した検索を実現しようすると、全グループに対して個々に構造指定検索を行ない、結果を統合しなければ、結果を得ることができない。

【0025】また、XML (Extended Markup Language) のように、必ずしも特定の構造を持たなくても良い構造化文書の規格化がW3C (World Wide Web Consortium) で進められており、SGMLのように特定のDTDに従った文書構造を持つ文書だけが検索対象とならないことが考えられる。

【0026】さらに、前記従来技術では、「タイトル」、「題目」のように、同じ意味（種別）を持つ構造であっても、要素型名が異なっていれば別の構造とみなされるため、「"タイトル"に"SGML"が含まれる文書」という構造指定検索では、「"題目"に"SGML"が含まれる文書」という条件を満たす文書は、検索結果として得ることができない。

【0027】特に、文書型定義が異なる場合、それぞれの文書型定義毎に、同じ種別の構造に異なる要素型名が

付けられることが考えられる。

【0028】例えば、「タイトル」に関する構造指定検索を行おうとすると、ユーザが「タイトル」、「題目」、「表題」、「TITLE」など、各文書型定義に出現する様々な「タイトル」を意味する要素型名を指定して構造指定検索条件を作成しなければ、必要とする文書をすべて取得することができない。また、登録文書の文書型定義をすべて知らなければ、ユーザが想定した要素型名で「タイトル」を意味する構造全てを網羅することはできない。例えば「T」という構造にタイトルを記述するといった文書型定義に従った文書は、それを知らないユーザからの構造指定検索では、決して取得することはできないであろう。

【0029】本発明のもう一つの目的は、上記問題点を解決し、文書構造が異なる文書が混在する文書群を、高精度でかつ効率的に構造指定検索する機能を提供することである。

【0030】さらに構造指定検索条件として、「章、節などのいずれかの項目のタイトルにおいて、“SGML”という単語を含む文書」を設定した場合、“タイトル”という構造条件を満たす構造を全て採さなければならないため、検索効率が悪くなるという問題がある。

【0031】検索条件として、「/文書/章/タイトル」のように、タイトルまでの全ての構造を最上位から順に指定するようにすれば、効率的に構造を特定することができる。しかし、ユーザが「/文書/章/タイトル」または「/文書/章/節/タイトル」または、、、のように、全ての構造を示した構造指定検索条件を作成しなければならないため、ユーザの負担が大きくなる上に、ユーザが検索対象の文書の構造を全ての把握しておかなければ、検索漏れが発生することになる。

【0032】本発明のさらにもう一つの目的は、上記問題点を解決し、複数の階層に出現する同じ種別の構造を指定した検索を、複雑な構造条件を指定することなく、効率的に実現する機能を提供することである。

【0033】

【課題を解決するための手段】上記課題を解決するため、本発明による文書登録および検索方法は、以下に示すステップを備える。

【0034】すなわち、本発明による文書登録方法において、文書の登録処理は、(1)登録対象文書の持つ論理構造を解析して解析済み文書データを生成し、これを文書データベースに登録する解析済み文書データ生成登録ステップと、(2)各登録対象文書の持つ論理構造を、登録順に従って順次重ね合わせ、文書中における出現位置および種別が同じである要素群は単一のメタ要素によって代表させ、文書中における出現位置が同じである文字列データ群は単一のメタ文字列データによって代表させることにより、メタ要素群およびメタ文字列データ群（以下、これらを総称してメタノードと呼ぶ）のメ

構造から構成される構造インデックスを生成し、該構造インデックスを構成するすべてのメタノードに対して、それらを構造インデックス中で一意に識別する識別子である文脈識別子を与える構造インデックス生成ステップと、(3)各登録対象文書について、その文書に対応する解析済み文書データ中に含まれるすべての文字列データと、その文字列データを構造インデックス中で代表するメタ文字列データの文脈識別子との対応関係の定義から構成される構造化全文データを生成する構造化全文データ生成ステップと、(4)各登録対象文書に対応する構造化全文データから、所定の部分文字列と、該部分文字列の前記登録対象文書中における文字位置情報と、前記登録対象文書を文書データベース中で一意に識別する文書識別子と、前記部分文字列を含む文字列データを構造インデックス中で代表するメタ文字列データの文脈識別子とを抽出し、前記文字位置情報および前記文書識別子および前記文脈識別子からなる構造化文字位置情報を生成し、前記部分文字列と前記構造化文字位置情報との対応関係を登録して文字列インデックスを更新する文字列インデックス更新ステップを有する。

【0035】また、本発明による文書検索方法において、登録済み文書の検索処理は、(1)前記構造インデックスを参照し、指定された構造条件を満たす文脈識別子の集合を決定する構造条件判定ステップと、(2)検索タームから所定の部分文字列を抽出し、前記文字列インデックスを参照して該部分文字列に対応する構造化文字位置情報の集合を抽出する構造化文字位置情報抽出ステップと、(3)前記構造化文字位置情報の集合中から、前記構造条件判定ステップで決定した集合に含まれる文脈識別子を持ち、かつ前記検索ターム上における部分文字列の並びと同じ位置関係を持つ構造化文字位置情報を抽出するインデックス検索ステップを有する。

【0036】さらに、本発明による文書検索方法において、複数の文書構造を持つ文書をまとめて登録する処理には、(1)構造インデックス中の複数の構造に出現する各構造の名称と構造の種別の対応を記述した種別定義テーブルを参照し、要素型名からその構造の種別を取得する種別取得ステップと、(2)文書の最上位構造と同じ種別を持つ最上位構造を持つ構造インデックスを取得する最上位構造判定ステップと、(3)構造化文書の登録時に、複数の文書構造を持つ文書の構造インデックスの上位に各構造インデックスをまとめる親ノード（ルートメタノード）を設けることにより、複数の構造インデックスを1つのメタ構造インデックスにまとめるルートメタノード作成ステップを有する。

【0037】もしくは、(1)構造インデックス中の複数の構造に出現する各構造の名称と構造の種別の対応を記述した種別定義テーブルを参照し、要素型名からその構造の種別を取得する種別取得ステップと、(4)登録文書を構造解析して得られる解析済み文書データに、全

文書に共通な仮の最上位構造を追加するステップを有する。

【0038】種別定義テーブルは、あらかじめ、人手により作成、あるいはシソーラスなどを利用して、同義語を同じ種別に割り当てるなどの方法により自動的に作成する。

【0039】さらに、本発明による文書検索方法において、構造インデクス中の多くの位置に出現する同じ種別の構造を指定した構造指定検索を効率的に実現するために、文書登録プログラムには、(1)文書登録時に構造インデクスを作成すると共に、別名構造インデクスを生成するステップを有する。

【0040】別名構造インデクスとは、例えば、作成日・更新日など、各文書構造毎に設定されうる情報を構造インデクスを辿らなくても、一括して検索できるように作成した構造インデクスである。別名構造インデクスで得られる種別を指定して構造指定検索することで、別名構造インデクスから、別名に対応する構造インデクス中の複数の構造を一括して得ることができるため、構造インデクスを辿って指定された構造の文脈識別子を得るよりも、効率的な検索を実現することが可能である。

【0041】

【発明の実施の形態】(第一の実施例)以下、本発明を適用した第一の実施例について、図面を用いて説明する。

【0042】はじめに、本実施例のシステム構成について説明する。

【0043】図1は、本発明による文書検索システムの第一の実施例の全体構成を示す図である。図1に示すとおり、本発明による文書検索システムの第一の実施例は、文書登録サブシステム101、文書検索サーバ102、文書検索クライアント103および104、ネットワーク105から構成される。

【0044】文書登録サブシステム101は、検索対象として入力される各文書の構造を解析し、検索時に必要となるインデックスデータを作成する。このインデックスデータは、ネットワーク105を介して文書検索サーバ102に転送され、後に文書検索サーバ102が構造検索処理を行う際に用いられる。

【0045】文書検索サーバ102は、検索クライアント103および104からの検索コマンドを受け取り、文書登録サブシステム101が作成したインデックスデータを用いて該検索コマンドの指定する条件に適合する文書内容の検索を行い、検索結果データを要求元の検索クライアントに送り返す。

【0046】検索クライアント103および104は、ユーザが対話的に検索条件を指定するための画面をディスプレイ上に表示し、この画面上でユーザが指定した検索条件を、文書検索サーバ102によって解釈可能な検索コマンドの形に変換し、この検索コマンドをネットワ

ーク105を介して文書検索サーバ102に送信する。前記のとおり文書検索サーバ102が検索コマンドに対応する検索処理を行い、検索結果データを送り返して来ると、検索クライアント102は受け取った検索結果データを画面に表示してユーザに提示する。なお、図1には2台のコンピュータ103および104を検索クライアントとして使用する構成例を示したが、検索クライアントは1台のみとする構成をとることも、3台以上とする構成をとることもできる。

【0047】ネットワーク105は、ローカルエリアネットワークおよび/または広域ネットワークであって、文書登録サブシステム101、文書検索サーバ102、検索クライアント103および104が各種データやコマンドを交換するために用いられる。

【0048】ここで、図1では文書登録サブシステム101から文書検索サーバ102にインデックスデータを転送するためにネットワーク105を使用するものとしたが、代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。あるいは、文書登録サブシステム101と文書検索サーバ102を1台のコンピュータ上に実装し、データ転送を行わない構成をとることもできる。

【0049】さらに、図1では検索クライアント103および104と文書検索サーバ102には別個のコンピュータを使用するものとしたが、1個以上の検索クライアントを文書検索サーバと同一のコンピュータ上で実行する構成をとることもできる。

【0050】以下、本実施例における文書登録サブシステム、すなわち図1の101について説明する。

【0051】図2は、本実施例における文書登録サブシステム101の構成を示す図である。

【0052】図2に示す文書登録サブシステム101は、ディスプレイ201、キーボード202、中央処理装置(CPU)203、フロッピーディスクドライブ204、フロッピーディスク205、通信制御装置206、主メモリ207、磁気ディスク装置208、システムバス209から構成される。

【0053】ディスプレイ201は、本サブシステムにおける処理の実行状況等を表示するために使用する。キーボード202は、文書登録処理の実行等を指示するコマンドを入力するために使用する。中央処理装置203は、本サブシステムを構成する各種プログラムを実行する。フロッピーディスクドライブ204は、フロッピーディスク205に対するデータの読み書きのために使用する。フロッピーディスク205は、予め登録対象文書を格納しておき、これを本サブシステムに入力するために使用する。通信制御装置206は、ネットワーク105を介して前記文書検索サーバ102と通信し、リクエストおよびデータの交換を行うために使用する。主メモリ207は、本サブシステムによる処理を実行するため

10

20

30

40

50

15

の各種プログラムおよび一時的なデータを保持するために使用する。磁気ディスク装置 208 は、登録された文書データおよび本サブシステムが生成するインデックスデータを格納するために使用する。システムバス 209 は、これらの各種装置を接続するために使用する。

【0054】主メモリ 207 中には、文書構造解析プログラム 210、構造インデックス作成プログラム 211、構造化全文データ生成プログラム 212、文字列インデックス作成プログラム 213、文書登録制御プログラム 214 およびシステムプログラム 215 が格納され、またワークエリア 216 が保持される。磁気ディスク装置 208 中には、解析済み文書データ格納領域 217、構造インデックス格納領域 218、構造化全文データ格納領域 219 および文字列インデックス格納領域 220 が確保される。

【0055】文書構造解析プログラム 210 は、SGML を用いて記述され、フロッピーディスク 205 に格納された登録対象文書を読みだし、その論理構造を解析して解析済み文書データを生成し、これを解析済み文書データ格納領域 217 に格納する。構造インデックス作成プログラム 211 は、前記解析済み文書データの持つ論理構造に関する情報を、構造インデックス格納領域 218 に格納されている構造インデックスに登録して構造インデックスを更新する。構造化全文データ生成プログラム 212 は、前記解析済み文書データから前記登録対象文書に関する構造化全文データを生成し、これを構造化全文データ格納領域 219 に格納する。文字列インデックス作成プログラム 213 は、前記構造化全文データから、所定の部分文字列と、該部分文字列の構造化文字位置情報との対応関係を表わすデータを生成し、これを文字列インデックス格納領域 220 に格納されている文字列インデックスに登録して文字列インデックスを更新する。

【0056】文書登録制御プログラム 214 は、文書構造解析プログラム 210、構造インデックス作成プログラム 211、構造化全文データ生成プログラム 212 および文字列インデックス作成プログラム 213 の起動および実行制御を行うとともに、これらのプログラムによって生成された解析済み文書データ、構造インデックスおよび文字列インデックスをネットワーク 105 を介して前記文書検索サーバ 102 に転送する。システムプログラム 215 は、周辺装置との間のデータの入出力など、コンピュータ上で本サブシステムを構成する各プログラムを実行するための基本機能を提供する。ワークエリア 216 は、プログラムの実行時に一時的に必要なデータを記憶するために用いられる。

【0057】なお、本実施例ではフロッピーディスク 205 に格納された登録対象文書を入力として読み込む構成としたが、光磁気ディスク、追記型光ディスク等、他の可搬型媒体から読み込む構成をとることもでき、ネ

16

ットワーク 105 を介して転送されてくる文書を入力とする構成をとることもできる。さらに、本実施例では生成された解析済み文書データ、構造インデックスおよび文字列インデックスを文書検索サーバ 102 に転送するためにネットワーク 105 を使用するものとしたが、代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。あるいは、文書登録サブシステム 101 と文書検索サーバ 102 を 1 台のコンピュータ上に実装し、データ転送を行わない構成をとることもできる。

【0058】次に、本実施例における文書登録処理の手順について説明する。

【0059】図 3 は、本発明の第一の実施例における文書登録処理の概略手順を示す PAD (Problem Analysis Diagram) 図である。キーボード 202 からの登録指示コマンド等により文書登録制御プログラム 214 が起動されると、本プログラムはまずフロッピーディスク 205 に格納されている登録対象文書の有無とその数を調べ、すべての登録対象文書について、ステップ 302 から 305 までに示す一連の処理を繰り返し実行する (ステップ 301)。

【0060】ステップ 302 では、フロッピーディスク 205 に格納されている登録対象文書群の中から、未処理の登録対象文書を 1 個選択して読み込む。ステップ 303 では、読み込んだ登録対象文書に文書識別子を割り当てる。文書識別子は、文書データベース中で特定の文書を一意に識別する番号である。

【0061】ステップ 304 では、この登録対象文書を入力として文書構造解析プログラム 210 を実行する。ここで、文書構造解析プログラム 210 は、この登録対象文書に対応する解析済み文書データを生成して解析済み文書データ格納領域 217 に格納する。

【0062】ステップ 305 では、ステップ 304 で生成した解析済み文書データを入力として構造インデックス作成プログラム 211 を実行する。構造インデックス作成プログラム 211 は、まず構造インデックス格納領域 217 から現時点での構造インデックスを読み出し、与えられた解析済み文書データの持つ構造情報をこの構造インデックスに登録し、更新された構造インデックスを再び構造インデックス格納領域 218 に格納する。

【0063】ステップ 306 では、ステップ 304 で生成した解析済み文書データを入力として、構造化全文データ生成プログラム 212 を実行する。構造化全文データ生成プログラム 212 は、与えられた解析済み文書データを参照して、ステップ 303 で読み込んだ登録対象文書に対応する構造化全文データを生成し、構造化全文データ格納領域 219 に格納する。

【0064】ステップ 307 では、ステップ 306 で生成した構造化全文データを入力として文字列インデックス作成プログラム 213 を実行する。文字列インデック

50

ス作成プログラム 2 1 3 は、まず文字列インデックス格納領域 2 2 0 から現時点での文字列インデックスを読み出し、構造化全文データから、所定の部分文字列と、該部分文字列の構造化文字位置情報との対応関係を表わすデータを生成してこれを文字列インデックスに登録し、更新された文字列インデックスを再び文字列インデックス格納領域 2 2 0 に格納する。

【0065】すべての登録対象文書についてステップ 3 0 2 から 3 0 7 までに示す一連の処理が終了すると、文書登録制御プログラム 2 1 4 はステップ 3 0 8 を実行して処理を終了する。ステップ 3 0 8 では、解析済み文書データ格納領域 2 1 7 に格納されたすべての解析済み文書データ、構造インデックス格納領域 2 1 8 に格納された構造インデックス、および文字列インデックス格納領域 2 2 0 に格納された文字列インデックスを、ネットワーク 1 0 5 を介して文書検索サーバ 1 0 2 に転送する。

【0066】次に、図 3 におけるステップ 3 0 4 の詳細、すなわち本実施例における文書構造解析プログラム 2 1 0 の処理手順について説明する。

【0067】ここで、文書構造解析プログラム 2 1 0 は、SGML を用いて記述された 1 個の登録対象文書を対象として構造解析処理を行う。SGML では、特定の種別に属する文書群に共通する論理構造を、DTD (Document Type Definition) によって定義する。図 4 が DTD の一例である。DTD は、文書を構成する論理要素（以下、これを単に「要素」と呼ぶ）の集合を定義することによって、文書の論理的構造を定義する。図 4 において、文字列 “<!ELEMENT” と文字列 “>” で囲まれた部分を要素型宣言と呼び、1 個の要素型宣言が、1 種類の要素型に属する要素群が共通して持つ名前（これを要素型名と呼ぶ）とその構造を規定する。要素型宣言中の左側に示されている文字列が要素型名、右側に示されている部分がその内容がとる構造の定義である。

【0068】図 4 に示す DTD において、要素型 “論文” に関する要素型宣言は、この要素型に属する要素の内容が、“タイトル”、“執筆者”、“日付”、“本文” および “文献リスト” という要素型に属する要素 1 個ずつを、この順序に従って並べた構造を持つことを規定している。ここで、複数の要素型名を文字 “,” で区切って並べることにより、それらの要素型名に属する要素が指定した順序で出現しなければならないことを表現する。

【0069】要素型 “執筆者” に関する要素型宣言は、この要素型に属する要素の内容が、要素型 “名前” に属する要素の 1 個以上の繰り返しからなる構造を持つことを規定している。ここで、要素型名の後ろに文字 “+” を付加することにより、その要素型名に属する要素が 1 個以上出現することを表現する。

【0070】要素型 “本文” に関する要素型宣言は、この要素型に属する要素の内容が、要素型 “章” に属する

要素の 0 個以上の繰り返しからなる構造を持つことを規定している。ここで、要素型名の後ろに文字 “*” を付加することにより、その要素型名に属する要素が 0 個以上出現することを表現する。

【0071】要素型 “章” に関する要素型宣言は、この要素型に属する要素の内容が、要素型 “章題” に属する要素 1 個の後ろに、要素型 “段落” または “備考” に属する要素を 0 個以上続け、さらにその後ろに要素型 “節” に属する要素を 0 個以上繰り返した構造を持つことを規定している。ここで、複数の要素型名を文字 “|” で区切って並べることにより、それらのいずれかの要素型に属する要素が出現することを表現する。

【0072】要素型 “節” に関する要素型宣言は、この要素型に属する要素の内容が、要素型 “節題” に属する要素 1 個の後ろに、要素型 “段落” または “備考” に属する要素を 0 個以上続け、さらにその後ろに要素型 “項” に属する要素を 0 個以上繰り返した構造を持つことを規定している。

【0073】要素型 “項” に関する要素型宣言は、この要素型に属する要素の内容が、要素型 “項題” に属する要素 1 個の後ろに、要素型 “段落” または “備考” に属する要素を 0 個以上繰り返した構造を持つことを規定している。

【0074】要素型 “文献リスト” に関する要素型宣言は、この要素型に属する要素の内容が、要素型 “文献” に属する要素の 1 個以上の繰り返しからなる構造を持つことを規定している。

【0075】要素型 “文献” に関する要素型宣言は、この要素型に属する要素の内容が、“タイトル”、“執筆者”、“日付” および “出典” という要素型に属する要素 1 個ずつを、この順序に並べた構造を持つことを規定している。

【0076】また、要素型 “タイトル”、“名前”、“日付”、“章題”、“節題”、“項題”、“強調” および “出典” に属する要素の内容は、単に “#PCDATA” と規定されている。これは、これらの要素がそれ以上の下位構造をもたず、単なる文字の列からなる内容を持つことを規定している。要素型 “段落” および “備考” に関する要素型宣言は、これらの要素型に属する要素が、要素型 “強調” に属する要素または単なる文字列を 0 個以上繰り返した構造を持つことを規定している。

【0077】DTD 中において、文字列 “<!ATTLIST” と文字列 “>” で囲まれた部分を属性リスト宣言と呼び、1 個の属性リスト宣言が、1 種類の要素型に属する要素群が共通して持つ属性を定義する。図 4 に示す DTD では、要素型 “備考” に属する要素が、属性 “type” を持つこと、この属性は “参照” または “注釈” のいずれかを値としてとることができ、指定が省略された場合には “参考” が値として与えられることを規定している。

【0078】図4に示すDTDに従って記述されたSGML文書の一例を、図5に示す。文書先頭の、文字列“<!DOCTYPE”と文字列“>”で囲まれた部分を文書型宣言と呼び、そのSGML文書が従うDTDと、最上位要素の要素型名を宣言する。図5に示した例では、この文書がファイル“ronbun.dtd”に格納されているDTDに従い、最上位要素の要素型名は“論文”であることが規定されている。ここでは、ファイル“ronbun.dtd”に図4に示した前記DTDが格納されているものとする。

【0079】図5に示すとおり、SGMLでは文書を構成する個々の要素について、その先頭位置と末尾位置を示すマークを付加することにより、文書構造を明示的に記述する。各要素の先頭位置を示すマークを“開始タグ”、終了位置を示すマークを“終了タグ”と呼ぶ。開始タグは、文字列“<”と“>”との間に、その要素の要素型名を記述することによって示す。終了タグは、文字列“</”と“>”との間に、その要素の要素型名を記述することによって示す。要素が属性を持つ場合、属性値の指定を開始タグ中（要素型名の後ろ）に記述することができる。属性値の指定は、属性名と属性値との間に文字列“=”を置くことによって示す。例えば図5において、開始タグ“<備考 type=注釈>”は、要素“備考”の属性“type”に属性値“注釈”を与えている。SGML文書中において、これらのタグを用いて文書構造を記述している部分を「文書インスタンス」と呼ぶ。

【0080】図3におけるステップ304の詳細、すなわち本実施例における文書構造解析プログラム210の処理手順を示すPAD図を、図7に示す。

【0081】図7に示すとおり、文書構造解析プログラム210は、SGMLを用いて記述された1個の登録対象文書を入力として起動されると、まず該文書の先頭に記述された文書型宣言を読み、その構文を解析する（ステップ701）。次に、ステップ702において、文書型宣言中における構文エラーの有無を判定する。構文エラーが検出された場合、ステップ703に進み、エラーメッセージを出力して処理を中断する。

【0082】文書型宣言に構文エラーがない場合にはステップ704に進み、その文書型宣言で指定されているDTDファイルが存在するかどうかを判定する。DTDファイルが検出されない場合、ステップ705に進み、エラーメッセージを出力して処理を中断する。

【0083】DTDファイルが検出された場合にはステップ706に進み、該ファイルの内容を読み込んでその構文解析を行う。次に、ステップ707において、DTD中における構文エラーの有無を判定する。構文エラーが検出された場合、ステップ708に進み、エラーメッセージを出力して処理を中断する。構文エラーが検出されなかった場合にはステップ709に進み、DTDが定義する文書構造モデルを記述したデータである文書構造テーブルをメモリ上に生成する。

【0084】次に、ステップ710において、前記文書構造テーブルを参照しながら文書インスタンスを読み込み、構造解析を行い、その結果として解析済み文書データを生成する。次に、ステップ711において、文書インスタンス中に構文エラーや構造エラー（DTDが定義する構造モデルからの逸脱）がないかどうかを判定する。構文エラーまたは構造エラーが検出された場合にはステップ712に進み、エラーメッセージを出力して処理を中断する。エラーが検出されなかった場合にはステップ713に進み、前記登録対象文書を識別する文書識別子と、前記ステップ710における構造解析によって得られた解析結果データとからなる解析済み文書データを、解析済み文書データ格納領域217に出力して処理を終了する。

【0085】ここで一例として、図5に示すSGML文書を登録対象文書として文書構造解析プログラム210を実行し、該文書が参照するDTDファイル“ronbun.dtd”の内容が図4に示すDTDだった場合について説明する。この場合、ステップ709において生成される文書構造テーブルは、図8に示すようなデータ構造をとる。図8に示すとおり、文書構造テーブルは構造定義と属性定義の二つの部分からなる。構造定義は、DTDを構成する各要素型の要素型名に対応させて、その要素型に属する要素がとりうる内容のデータモデルを定義する。また属性定義は、DTDを構成する各要素型の要素型名に対応させて、その要素型に属する要素が持つ各属性の属性名、属性値のタイプおよびデフォルト値を定義する。この構造定義を参照することにより、文書インスタンス中に現れる要素群の並び順や階層関係が正しいかどうか（構造エラーの有無）を判断し、また省略されたタグや属性値指定があった場合、それらを補足することができる。

【0086】図5に示すSGML文書が登録対象文書として文書構造解析プログラム210への入力として与えられ、そのDTDが図4に示すものだった場合、図6に示す木構造データが解析済み文書データとして得られる。図6は、図5に示すSGML記述によって表現されている文書の論理構造を、図形的に示した模式図である。図6に示すように、構造化文書の持つ論理的構造は、個々の要素を中間節、文字列データを終端節とする木構造としてとらえることができる。図6では、各要素を楕円形、文字列データを矩形で表わしている。

【0087】なお、本実施例ではSGMLを用いて記述された構造化文書を登録対象文書として処理する構成をとったが、ODA（Open Document Architecture）など、他の形式によって記述された構造化文書を登録対象文書とするように構成することもできる。

【0088】図9は、図3におけるステップ305の詳細、すなわち本実施例における構造インデックス作成プログラム211の処理手順を示すPAD図である。

【0089】構造インデックス作成プログラム211は、まずステップ901において、既存の構造インデックスが構造インデックス格納領域218中に存在しているかどうかを判定する。該領域中に構造インデックスが存在しない場合にはステップ902に進み、初期状態（空）の構造インデックスを生成する。既存の構造インデックスを検出した場合にはステップ903に進み、該構造インデックスを読み込む。

【0090】次に、ステップ904において、解析済み文書データ格納領域217から登録対象文書の解析済み文書データを読み込む。

【0091】次に、ステップ905において、前記解析済み文書データの木構造を構成するすべてのノード（要素および文字列データ）を対象として、ステップ906からステップ909に示す処理を繰り返す。

【0092】ステップ906では、解析済み文書データ中で現在着目しているノードについて、構造インデックス中に、該ノードに対応するメタノード（メタ要素またはメタ文字列データ）が存在するかどうかを判定する。対応メタノードが存在しない場合はステップ907に進み、該ノードに対応するメタノードを生成して構造インデックスに登録し、さらに該登録したメタノードに対して、該メタノードを構造インデックス中で一意に識別する番号である文脈識別子を割り当てる（ステップ908）。ステップ909では、解析済み文書データ中で現在着目しているノードと、構造インデックス中で該ノードに対応するメタノードを識別する文脈識別子との対応関係を、解析済み文書データに付加して該解析済み文書データを更新する。

【0093】ステップ905以下の繰り返しを終了するとステップ910に進み、前記更新された解析済み文書データを出力して解析済み文書データ格納領域217に格納する。次に、ステップ911では、前記更新された構造インデックスを出力して構造インデックス格納領域218に格納し、処理を終了する。

【0094】ここで、前記ステップ905において解析済み文書データの木構造を構成するすべてのノードを対象とする繰り返し処理を行う際に、該木構造を辿り、個々のノードを処理していく順序を図10を用いて説明する。図10では、楕円形で要素ノード、矩形で文字列ノードを表現し、あるノードが複数の下位ノードを持つ場合には、各下位ノードをその出現順に左から右に並べて表現している。また、各ノードに表示された数字がその処理順序を示している。図10に示すとおり、ステップ905においてノード群を処理していく順序は、木構造の根に位置するノードから出発し、ある特定のノードとその下位ノード群を処理する際には、まず該ノードの処理を行い、次に該ノードの下位ノード群を、その出現順に従って処理していく順序となる。

【0095】次に、前記ステップ906における処理、

すなわち解析済み文書データ中で現在着目しているノードについて、構造インデックス中に、該ノードに対応するメタノードが存在するかどうかを判定する処理の内容を、図11を用いて説明する。図11は、図の左側に示す解析済み文書データの木構造を構成するノード群と、右側に示す構造インデックスの木構造を構成するノード（メタノード）群との対応関係を示す図である。

【0096】ここで、本実施例では、解析済み文書の木構造中におけるあるノードの木構造アドレスと、構造インデックスの木構造中におけるあるメタノードの木構造アドレスとが等しい場合に、該ノードと該メタノードとは対応するものと定義する。ここでいう木構造アドレスとは、木構造の根から出発して上位ノードから下位ノードへと辿り、特定のノードに至るまでの道筋を、該道筋上に存在する各ノードの種別（要素か文字列データか、要素の場合、どの要素型に属するか）と、それらのノードが種別の等しい兄弟ノード群中で何番目に現れたかを示す番号との組み合わせによって表現するアドレスのことである。

【0097】例えば、図11に示す解析済み文書データ中のノード群のうち、ノード1101は上位ノードをもたず、兄弟ノード中では最初の“論文”要素ノードであるから、その木構造アドレスは“/論文[1]”と表記することができる。同様に、ノード1102はノード1101の下位ノードであり、兄弟ノード中では最初の“章”要素ノードであるから、その木構造アドレスは“/論文[1]/章[1]”と表記することができる。また、ノード1103はノード1102の下位ノードであり、兄弟ノード中では2番目の“節”要素ノードであるから、その木構造アドレスは“/論文[1]/章[1]/節[2]”と表記することができる。また、ノード1104はノード1103の下位ノードであり、兄弟ノード中では最初の“段落”要素ノードであるから、その木構造アドレスは“/論文[1]/章[1]/節[2]/段落[1]”と表記することができる。

【0098】同様にして、図11右側の構造インデックスの木構造を構成する各メタノードについてその木構造アドレスを求めてみると、メタノード1105の木構造アドレスは“/論文[1]”となり、ノード1101の木構造アドレスと等しくなる。同様に、メタノード1106の木構造アドレスは“/論文[1]/章[1]”となってノード1102の木構造アドレスと等しく、メタノード1107の木構造アドレスは“/論文[1]/章[1]/節[2]”となってノード1103の木構造アドレスと等しくなる。よって、前記ステップ906において、ノード1101はメタノード1105と、ノード1102はメタノード1106と、ノード1103はメタノード1107とそれぞれ対応するものと判定される。

【0099】なお、図11の構造インデックス中にはノード1104と等しい木構造アドレスを持つメタノードはないので、ノード1104と対応するメタノードは構

23

造インデックス中に存在しないものと判定され、前記ステップ907において新たなメタノードが生成され、該メタノードが構造インデックスに登録されることになる。前記ステップ907において、あるノードに対応する新たなメタノードに登録する際には、該ノードの上位ノードに対応するメタノードの持つ下位ノード群の末尾に、該ノードに対応する種別のメタノードを追加する。図11のノード1104に対応するメタノードに登録する場合には、ノード1104の上位ノードであるノード1103に対応するメタノード1107の下位に、要素型“段落”のメタノードが追加され、該メタノードは兄弟メタノード群の末尾に置かれることになる。

【0100】次に、複数の解析済み文書データを順次重ね合わせるにより構造インデックスを生成していく過程について、図12を用いて説明する。図12において、1201、1203および1205は、それぞれ登録対象文書の解析済み文書データを表わしている。これらの解析済み文書データの構造を既存の構造インデックス上に順次重ね合わせるにより、構造インデックスが形成されていく。まず最初に文書1の解析済み文書データ1201が入力されると、最初の段階では構造インデックスは初期状態（空）であるため、該解析済みデータと等価な木構造が生成されてそのまま構造インデックスに登録され、構造インデックスは1202に示す状態となる。新たに生成されたメタ要素にはE1からE5までの文脈識別子、新たに生成されたメタ文字列データにはC1からC3までの文脈識別子が割り当てられる。

【0101】次に文書2の解析済み文書データ1203が入力されると、既存の構造インデックス（1202）と構造が重複する部分については何も行わず、1202上に対応する部分がなかった部分構造（図中の斜線部分）だけが新たに登録される。新たに生成されたメタ要素には文脈識別子E6およびE7、新たに生成されたメタ文字列データには文脈識別子C4が割り当てられる。次に文書3の解析済み文書データ1205が入力されると、既存の構造インデックス（1204）と構造が重複する部分については何も行わず、1204上に対応する部分がなかった部分構造（図中の斜線部分）だけが新たに登録される。新たに生成されたメタ要素には文脈識別子E8、E9およびE10、新たに生成されたメタ文字列データには文脈識別子C5およびC6が割り当てられる。このようにして、3個の文書が登録された段階で、構造インデックスは1206に示す状態となる。

【0102】図13は、図3におけるステップ306の詳細、すなわち本実施例における構造化全文データ生成プログラム212の処理手順を示すPAD図である。

【0103】構造化全文データ生成プログラム212は、まずステップ1301において、解析済み文書データ格納領域217から前記登録対象文書の解析済み文書データを読み込む。

24

【0104】次に、ステップ1302において、前記登録対象文書を識別する文書識別子を構造化全文データ格納領域219に出力する。

【0105】次に、ステップ1303において、前記解析済み文書データの木構造を構成するすべてのノード（要素ノードおよび文字列データノード）を対象として、ステップ1304からステップ1306に示す処理を繰り返す。

【0106】ステップ1304では、解析済み文書データ中で現在着目しているノードについて、該ノードが要素ノードであるか文字列データノードであるかを判定し、該ノードが文字列データノードである場合に限りステップ1305に進む。ステップ1305では、前記解析済み文書データから、前記現在着目している文字列データノードに対応する文脈識別子を求め、該文脈識別子を構造化全文データ格納領域219に出力する。次に、ステップ1306で、前記現在着目している文字列データノードの内容文字列を構造化全文データ格納領域219に出力する。

【0107】ステップ1303以下の繰り返しをすべて終了した段階で本プログラムは処理を終了する。

【0108】図14に、構造化全文データ生成プログラム212が出力する構造化全文データのファイル形式を示す。図14は、図5に示したSGML文書を入力として構造化全文データを生成した場合について例示している。図14に示すとおり、本実施例における構造化全文データのデータファイルは、先頭に文書識別子を記述し、その後ろに、文脈識別子と該文脈識別子に対応する内容文字列との対を、文書中に存在する文字列データの数だけ繰り返した構造をとる。

【0109】例えば、図14に示す構造化全文データに対応する登録対象文書の文書識別子は“D1”であり、図5において“日付”要素の内容として記述されていた文字列データには文脈識別子“C5”が与えられている。なお、図14他ではこれらの識別子を記号により表現しているが、文書識別子としてデータ中に実際に記録される値は登録対象文書の集合中で特定の文書を一意に識別する番号（整数値）であり、文脈識別子の値は構造インデックスを構成するメタノードの集合中で特定のメタノードを一意に識別する番号（整数値）である。

【0110】図15は、図3におけるステップ307の詳細、すなわち本実施例における文字列インデックス作成プログラム213の処理手順を示すPAD図である。

【0111】文字列インデックス作成プログラム213は、まずステップ1501において、既存の文字列インデックスが文字列インデックス格納領域220中に存在しているかどうかを判定する。該領域中に文字列インデックスが存在しない場合にはステップ1502に進み、初期状態（空）の文字列インデックスを生成する。既存の文字列インデックスを検出した場合にはステップ15

03に進み、該文字列インデックスを読み込む。

【0112】次に、ステップ1504において、登録対象文書の構造化全文データを構造化全文データ格納領域219から読み込む。

【0113】次に、ステップ1505において、前記構造化全文データを構成するすべての内容文字列を対象として、ステップ1506から1507に示す処理を繰り返す。

【0114】ステップ1506では、構造化全文データ中で現在着目している内容文字列から、所定の部分文字列を抽出する。ステップ1507では、前記ステップ1506で抽出した各部分文字列と該部分文字列の構造化文字位置情報との対応関係を、前記文字列インデックスに登録する。

【0115】ステップ1505以下の繰り返しを終了するとステップ1508に進み、不要となった構造化全文データを構造化全文データ格納領域219から削除して破棄する。次に、ステップ1509において、更新された文字列インデックスを出力して文字列インデックス格納領域220に格納し、処理を終了する。

【0116】ここで、前記ステップ1506において、ある内容文字列から所定の部分文字列を抽出する際には、予め抽出すべき部分文字列の長さを定めておき、対象とする内容文字列の先頭から出発して、開始位置を1ずつ増しながら前記予め定めておいた長さの部分文字列を順次抽出する。例えば、抽出する部分文字列の長さを2文字とし、処理対象として図14に示す内容文字列群のうち文脈識別子C129に対応する内容文字列“変換処理の実例”を用いた場合、抽出される部分文字列は“変換”、“換処”、“処理”、“理の”、“の実”および“実例”の6個となる。

【0117】更に、内容文字列の末尾部分については、長さ1文字から部分文字列長-1文字までの各部分文字列をも抽出する。前記の例では“例”が抽出される。これらの部分文字列を前記ステップ1507において文字列インデックスに登録する際には、各部分文字列と、それらが出現した位置を表わす構造化文字位置情報との対応関係を登録する。ここで、構造化文字位置情報は、対応する部分文字列を含む文書の文書識別子、該文書中において前記部分文字列を含む文字列データの文書構造中における位置を識別する文脈識別子、および文書中における前記部分文字列の先頭文字位置から構成される。

【0118】図16に、本実施例における文字列インデックスのデータ構造を示す。図16は、文字列インデックス作成プログラム213を用いて図14に示す構造化全文データを処理し、文字列インデックスに該構造化全文データに含まれる部分文字列群の登録を終えた段階において、該文字列インデックスのデータ構造の一部（前記内容文字列“変換処理の実例”に関連する部分）を図示したものである。ただし、図16では、前記内容文字

列の末尾“例”に対応する文字ノードおよび構造化文字位置情報は省略している。また、前記内容文字列の直前に位置する文字の文字位置を“X”として、相対的に文字位置を表現している。

【0119】図16に示すとおり、文字列インデックスは、登録対象文書中に現れる所定長さの部分文字列すべてについて、その出現位置情報（文書識別子、文脈識別子および先頭文字位置の組み合わせからなる構造化文字位置情報）のリストを保持する。ここで、インデックス検索を高速化するため、1文字目が共通する部分文字列群については、1文字目の情報をすべての部分文字列が共有するデータ構造をとり、また文字列インデックスのルートから1文字目のノードへのポインタの並びは、ポインタが指す文字の文字コード順となるように配列する。同様に、1文字目のノードから2文字目のノードへのポインタの並びも、ポインタが指す文字の文字コード順となるように配列する。

【0120】文書データベースに登録するすべての登録対象文書を処理してその内部に現れる部分文字列群を文字列インデックスに登録しておけば、この文字列インデックスを参照するだけで（文書データ本体を一切走査することなく）任意の2文字からなる文字列がどの文書中のどの位置に出現するかを知ることができる。（2文字以外の長さの文字列を検索する方法については後述する。）なお、本実施例では、予め定める部分文字列の長さを2文字としたが、該長さをこれ以外の値としても同様な文字列インデックスを構築することができる。また、本実施例では部分文字列の長さを固定長としたが、この長さを可変として同様な文字列インデックスを構築することもできる。

【0121】以上が、本実施例における文書登録サブシステム101の説明である。

【0122】以下、本発明の第一の実施例における文書検索サーバ、すなわち図1の102について説明する。

【0123】図17は、本実施例における文書検索サーバ102の構成を示す図である。

【0124】図17に示す文書検索サーバ102は、ディスプレイ201、キーボード202、中央処理装置（CPU）203、通信制御装置206、主メモリ207、磁気ディスク装置208、システムバス209から構成される。

【0125】ディスプレイ201は、本サーバの稼動状況等を表示するために使用する。キーボード202は、本サーバの起動・停止等を指示するコマンドを入力するために使用する。中央処理装置203は、本サーバを構成する各種プログラムを実行する。通信制御装置206は、ネットワーク105を介して前記文書登録サブシステム101および前記検索クライアント（103および104）と通信し、リクエストおよびデータの交換を行うために使用する。主メモリ207は、本サーバによる

処理を実行するための各種プログラムおよび一時的なデータを保持するために使用する。磁気ディスク装置208は、文書データベースを構成する文書データ群および本サーバが文書検索時に参照するインデックスデータを格納するために使用する。システムバス209は、これらの各種装置を接続するために使用する。

【0126】主メモリ207中には、検索条件解析プログラム1701、文字列インデックス検索プログラム1702、文書検索制御プログラム1703およびシステムプログラム215が格納され、またワークエリア216が保持される。磁気ディスク装置208中には、解析済み文書データ格納領域217、構造インデックス格納領域218、文字列インデックス格納領域220および検索結果データ格納領域1704が確保される。

【0127】検索条件解析プログラム1701は、検索クライアント103および104から受信した検索リクエスト中に含まれる検索条件式を解析し、文字列インデックス検索プログラム1702によって直接検索可能な条件指定に翻訳する。文字列インデックス検索プログラム1702は、検索条件解析プログラム1701によって翻訳された条件指定に従って、文字列インデックス格納領域220に格納されている文字列インデックスを検索し、得られた検索結果データを検索結果データ格納領域1704に格納する。

【0128】文書検索制御プログラム1703は、検索条件解析プログラム1701および文字列インデックス検索プログラム1702の起動および実行制御を行うとともに、ネットワーク105を介して、文書登録サブシステム101および検索クライアント(103および104)との間でリクエストおよびデータの交換を行う。システムプログラム215は、周辺装置との間のデータの入出力など、コンピュータ上で本サーバを構成する各プログラムを実行するための基本機能を提供する。ワークエリア216は、プログラムの実行時に一時的に必要なデータを記憶するために用いられる。

【0129】なお、本実施例では文書登録サブシステム101および検索クライアント103および104との間でデータを転送するためにネットワーク105を使用するものとしたが、代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。また、文書登録サブシステム101と文書検索サーバ102を1台のコンピュータ上に実装し、これらの間でデータ転送を行わない構成をとることもできる。また、1個以上の検索クライアントを文書検索サーバ102と同一のコンピュータ上で実行し、これらの間でデータ転送を行わない構成をとることもできる。

【0130】図18は、本発明の第一の実施例における文書検索処理の概略手順を示すPAD図である。キーボード202からのサーバ起動コマンド等により文書検索

制御プログラム1703が起動されると、本プログラムはサーバとして文書登録サブシステム101および検索クライアント(103、104等)からリクエストを受信してはその処理を行うループに入る(ステップ1801)。このループは、キーボード202からサーバの停止を指示するコマンドが入力されるまで継続する。

【0131】ステップ1801のループは、文書登録サブシステム101および検索クライアント(103および104)からリクエストを受信する処理(ステップ1802)と、受信したリクエストの種別を判定し、該種別に対応する処理に分岐する処理(ステップ1803)を繰り返す。

【0132】ステップ1803では、受信したリクエストの種別を判定し、該リクエストが、文書登録サブシステム101から送信されたデータベース更新リクエスト(新たな文書群を登録して文書データベースを更新することを求めるリクエスト)であった場合、ステップ1804およびステップ1805からなる処理に分岐する。

【0133】また、前記リクエストが、検索クライアント(103、104等)から送信された文書検索リクエスト(特定の検索条件を満たす文書群の検索を求めるリクエスト)であった場合、ステップ1806、1807および1808からなる処理に分岐する。また、前記リクエストが、検索クライアント(103、104等)から送信された、検索結果問い合わせリクエスト(特定の検索処理の結果を問い合わせるリクエスト)であった場合、ステップ1809からなる処理に分岐する。また、前記リクエストが、検索クライアント(103、104等)から送信された、文書転送リクエスト(指定された文書データの転送を求めるリクエスト)であった場合、ステップ1810からなる処理に分岐する。分岐先の処理が終了した後は再びステップ1802に戻ってループを継続する。

【0134】ステップ1804では、文書登録サブシステム101から、新規に登録された文書群の解析済み文書データを受信し、該解析済み文書データを解析済み文書データ格納領域216に追加する。次に、ステップ1805では、文書登録サブシステム101から、新規に登録された前記文書群の内容を反映して更新された構造インデックスおよび文字列インデックスを受信し、これらをそれぞれ構造インデックス格納領域218および文字列インデックス格納領域220に格納する。

【0135】ステップ1806では、検索条件解析プログラム1701を実行し、文書検索リクエスト中で指定されている検索条件を解析し、該検索条件を、文字列インデックス検索プログラム1702によって直接処理可能な条件指定(以下、これを展開済み検索条件データと呼ぶ)に変換する。次に、ステップ1807では、前記ステップ1806によって生成された展開済み検索条件データを入力として文字列インデックス検索プログラム

1702を実行し、該展開済み検索条件データが指定する条件を満たす文書群を検索して、検索結果データを求め、該検索結果データを一意に識別する検索結果識別子と対応付けて検索結果データ格納領域1704に格納する。次に、ステップ1808では、前記検索結果識別子を、要求元の検索クライアントに返送する。

【0136】ステップ1809では、問い合わせの内容に応じて前記ステップ1807で求めた検索結果データの一部または全体を検索結果データ格納領域1704から抽出し、要求元の検索クライアントに転送する。

【0137】ステップ1810では、文書転送リクエスト中で指定されている文書（複数の文書が指定されている場合には指定されている文書すべて）の解析済み文書データを解析済み文書データ格納領域217から抽出し、要求元の検索クライアントに転送する。

【0138】図19は、図18におけるステップ1806の詳細、すなわち本実施例における検索条件解析プログラム1701の処理手順を示すPAD図である。

【0139】検索条件解析プログラム1701は、文書検索リクエスト中で指定されている検索条件を入力として起動されると、まずステップ1901において、該検索条件中に構造条件が含まれているかどうかを判定する。そして、構造条件が含まれている場合に限り、ステップ1902およびステップ1903からなる処理を実行する。構造条件が含まれていなかった場合にはステップ1904に進む。

【0140】ステップ1902では、構造インデックス格納領域218から構造インデックスを読み込む。次に、ステップ1903では、該構造インデックスを参照して、前記構造条件を満たす構造内に含まれるすべての文字列データの文脈識別子の集合を求める。以下、該集合を文脈識別子集合と呼ぶ。

【0141】ステップ1904では、前記検索条件中に文字列条件として指定された文字列の長さが、前記文字列インデックスを作成する際に予め定めた部分文字列長を超えているかどうかを判定する。前記指定文字列の長さが前記部分文字列長を超えている場合にはステップ1905に進み、前記指定文字列の先頭から、開始文字位置を1ずつ増しつつ、前記部分文字列長に等しい長さの部分文字列群を抽出し、これらの部分文字列を要素とする部分文字列リストを生成する。前記指定文字列の長さが前記部分文字列長を超えていない場合にはステップ1906に進み、空の（要素をもたない）部分文字列リストを生成する。

【0142】ステップ1907では、前記ステップ1903で求めた文脈識別子集合、前記検索条件中に含まれていた指定文字列、および前記ステップ1905またはステップ1906で生成した部分文字列リストから構成される展開済み部分文字列データを生成して処理を終了する。

【0143】ここで、図20は、検索条件解析プログラム1701の処理過程における、展開済み解析条件データの生成例を示す図である。

【0144】図20において、2001は、文書検索リクエスト中で指定された検索条件の一例である。検索条件2001は、構造条件指定“章/段落[1]”と、文字列条件指定“ガード”とから構成されている。前記検索条件は、“章”要素の直接の下位にある最初の“段落”要素内に、文字列“ガード”が出現するケースを検索すべきことを指定している。

【0145】ここで、構造インデックスの内容が2002に示すとおりであったとすると、前記ステップ1903においてこの構造インデックスを参照することにより、前記構造条件指定を満たす“段落”要素の文脈識別子はE5およびE14であることがわかる。従って、これらの段落の下位にある文字列データ、すなわち文脈識別子がC3またはC9である文字列データ内に、文字列“ガード”が出現するケースを検索すればよいことがわかる。ただし、検索に用いる文字列インデックスには、長さ2の部分文字列についてののみその出現位置が登録されているので、3文字からなる前記指定文字列を直接検索することはできない。そこで、ステップ1905において、前記指定文字列を分解して長さ2の部分文字列からなるリストを生成する。前記のとおり指定文字列が“ガード”だった場合、抽出される部分文字列は“ガー”および“ード”となる。

【0146】この結果、前記ステップ1907において、2003に示す展開済み検索条件データ、すなわち文脈識別子集合が{C3, C9}、指定文字列が“ガード”、部分文字列リストが{“ガー”, “ード”}であるデータが生成される。

【0147】図21は、図18におけるステップ1807の詳細、すなわち本実施例における文字列インデックス検索プログラム1702の処理手順を示すPAD図である。

【0148】文字列インデックス検索プログラム1702は、前記検索条件解析プログラム1701が生成した展開済み検索条件データを入力として起動される。本プログラムは、起動されるとまずステップ2101において、文字列インデックス格納領域220から文字列インデックスを読み込む。次に、ステップ2102に進み、検索結果データを初期化する。

【0149】次に、ステップ2103では、前記展開済み検索条件データ中に含まれている指定文字列の長さ、と、前記文字列インデックスを作成する際に予め定めた部分文字列の長さとを比較する。前記指定文字列の長さが前記部分文字列の長さに等しい場合にはステップ2104に分岐する。前記指定文字列の長さが前記部分文字列の長さに満たない場合にはステップ2105に分岐する。前記指定文字列の長さが前記部分文字列の長さを超

10

20

30

40

50

える場合にはステップ2106に分岐する。

【0150】ステップ2104では、文字列インデックス中で前記指定文字列を検索し、該文字列に対応している構造化文字位置の集合を求め、次に該集合の中から、前記展開済み検索条件データ中の文脈識別子集合に含まれているいずれかの文脈識別子を持つ構造化文字位置情報群のみを抽出し、該抽出した構造化文字位置情報群からなるヒット位置集合を生成する。

【0151】ステップ2105では、文字列インデックス中で前記指定文字列を検索し、該文字列の末尾に対応する文字ノードより前方に存在するすべての構造化文字位置情報の集合を求め、次に該集合の中から、前記展開済み検索条件データ中の文脈識別子集合に含まれているいずれかの文脈識別子を持つ構造化文字位置情報群のみを抽出し、該抽出した構造化文字位置情報群からなるヒット位置集合を生成する。

【0152】ステップ2106では、前記展開済み検索条件データ中の部分文字列リストを構成する各部分文字列について、ステップ2107を繰り返す。ステップ2107では、文字列インデックス中で前記部分文字列を検索し、該文字列に対応している構造化文字位置情報の集合を求め、次に該集合の中から、前記展開済み検索条件データ中の文脈識別子集合に含まれているいずれかの文脈識別子を持つ構造化文字位置情報群のみを抽出し、該抽出した構造化文字位置情報群を前記部分文字列に対応付けて記憶する。

【0153】ステップ2106での繰り返しを終了するとステップ2108に進み、前記ステップ2107で対応付けを行った構造化文字位置情報群について接続判定処理を行い、連続した文字列として前記指定文字列を構成する構造化文字位置情報のグループのみを抽出し、該抽出した各グループ中で前記指定文字列の先頭に位置する部分文字列に対応する構造化文字位置情報のみを抽出し、該抽出した構造化文字位置情報群からなるヒット位置集合を生成する。

【0154】ステップ2103から分岐したすべての処理を終えたとステップ2109に進み、前記ヒット位置集合中に含まれる構造化文字位置情報群を、同一の文書識別子を持つものからなるグループにまとめて前記検索結果データに登録する。

【0155】ここで、前記ステップ2108、すなわち文字列インデックス検索プログラム1702の処理過程における接続判定処理について、図22を用いてさらに詳しく説明する。

【0156】図22において、2201は、文字列インデックスの一例（部分）を表わしている。2201に示すデータを保持している文字列インデックス上で、図20の2003に示す展開済み検索条件データが示す条件に従って検索を行うと、前記ステップ2107に示したとおり、まず部分文字列“ガー”および“ード”に対応

する構造化文字位置情報群の中から、文脈識別子がC3またはC9であるもののみが抽出される。該抽出された構造化文字位置情報群を、部分文字列に対応付けたデータを2202に示す。接続判定処理は該データの上で行われる。

【0157】前記ステップ2108に示す接続判定処理では、抽出された構造化文字位置情報群の中に、全体として前記指定文字列を構成する一連の構造化文字位置情報の組み合わせが存在するかどうかを判定する。ここで、このような組み合わせは次に示す条件を満たさなければならない。

【0158】（1）構造化文字位置情報群の間で、文書識別子がすべて一致している。

【0159】（2）構造化文字位置情報群の間で、文脈識別子がすべて一致している。

【0160】（3）構造化文字位置情報を、その文字位置の値が小さいものから順に並べ、それらの文字位置に従って対応する部分文字列群を配置すると、全体として指定文字列に等しい文字列が得られる。

【0161】ここで、2202に示す事例には、全体として指定文字列“ガード”を構成する組み合わせが一例含まれている。

【0162】上記の条件を満たす構造化文字位置情報の組み合わせが見つかったと、該組み合わせ中に含まれる構造化文字位置情報群のうち、文字位置の値が最小であるものを代表として選び、前記ヒット位置集合に登録する。

【0163】次に、図23は、個々の検索処理の結果として生成される検索結果データのデータ構造を示す図である。本図に示すとおり、検索結果データは、ヒット位置集合中に含まれる文字位置情報群を文書識別子ごとにグループ化し、それらのグループを要素とするリストを作り、さらに検出文書の総数を示す情報を付加した構造を持つ。検索結果データは、検索結果データの集合中でその検索結果データを一意に識別する検索結果識別子と対応付けられたうえで、検索結果データ格納領域1704に格納される。

【0164】次に、図18におけるステップ1809、すなわち検索結果問い合わせリクエストの内容に対応して検索結果を要求元クライアントに転送する処理について、図24を用いてさらに詳しく説明する。図24は、前記ステップ1809の詳細な処理手順を示すPAD図である。

【0165】ここで、検索結果問い合わせリクエストの本体は、検索結果識別子指定、問い合わせ種別指定および文書識別子指定の三部分からなる。問い合わせの種別によっては、文書識別子指定をもたない場合もある。

【0166】図24に示すとおり、前記ステップ1809に対応する処理では、まずステップ2401において、問い合わせリクエスト中で指定されている検索結果

識別子に対応する検索結果データを検索し、該検索結果データを検索結果データ格納領域1704から読み込む。

【0167】次に、ステップ2402では、問い合わせ種別を判定し、該問い合わせ種別が検出文書数問い合わせであった場合にはステップ2403、該問い合わせ種別が文書識別子問い合わせであった場合にはステップ2404、該問い合わせ種別が文字位置情報問い合わせであった場合にはステップ2405に分岐する。

【0168】ステップ2403では、前記ステップ2401で読み込んだ検索結果データから検出文書数を抽出し、該検出文書数の値を要求元の検索クライアントに転送して処理を終了する。

【0169】ステップ2404では、前記ステップ2401で読み込んだ検索結果データ中に含まれるすべての文書識別子からなる集合を求め、該集合を要求元の検索クライアントに転送して処理を終了する。

【0170】ステップ2405では、前記ステップ2401で読み込んだ検索結果データから、問い合わせ中で指定された文書識別子に対応する構造化文字位置情報のリストを抽出し、該リストを要求元の検索クライアントに転送して処理を終了する。以上が、本実施例における文書検索サーバ102の説明である。

【0171】以下、本発明の第一の実施例における文書検索クライアント、すなわち図1の103および104について説明する。

【0172】図25は、103および104に共通する、本実施例における文書検索クライアントの構成を示す図である。

【0173】図25に示す文書検索クライアントは、ディスプレイ201、キーボード202、中央処理装置(CPU)203、通信制御装置206、主メモリ207、磁気ディスク装置208、システムバス209から構成される。

【0174】ディスプレイ201は、ユーザが対話的に検索条件を入力するための画面や検索結果等を表示するために使用する。キーボード202は、検索条件、検索処理の実行等を指示するコマンドを入力するために使用する。中央処理装置203は、本クライアントを構成する各種プログラムを実行する。通信制御装置206は、ネットワーク105を介して前記文書検索サーバ102と通信し、リクエストおよびデータの交換を行うために使用する。主メモリ207は、本クライアントによる処理を実行するための各種プログラムおよび一時的なデータを保持するために使用する。磁気ディスク装置208は、検索結果として得られた文書およびその他のデータを格納するために使用する。システムバス209は、これらの各種装置を接続するために使用する。

【0175】主メモリ207中には、検索条件入力プログラム2501、検索結果表示プログラム2502、ク

ライアント制御プログラム2503およびシステムプログラム215が格納され、またワークエリア216が保持される。磁気ディスク装置208中には、解析済み文書データ格納領域217および検索結果データ格納領域1704が確保される。

【0176】検索条件入力プログラム2501は、ユーザと対話しつつ検索条件の入力および解釈を行う。検索結果表示プログラム2502は、文書検索サーバ102から受け取った検索結果の表示を行う。クライアント制御プログラム2503は、検索条件入力プログラム2501および検索結果表示プログラム2502の起動および実行制御を行うとともに、ネットワーク105を介して、文書検索サーバ102との間でリクエストおよびデータの交換を行う。システムプログラム215は、周辺装置との間のデータの入出力など、コンピュータ上で本クライアントを構成する各プログラムを実行するための基本機能を提供する。ワークエリア216は、プログラムの実行時に一時的に必要となるデータを記憶するために用いられる。

【0177】なお、本実施例では文書検索サーバ102との間でデータを転送するためにネットワーク105を使用するものとしたが、代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。また、1個以上の検索クライアントを文書検索サーバ102と同一のコンピュータ上で実行し、これらの間でデータ転送を行わない構成をとることもできる。本クライアントにプリンタを接続し、検索結果を印刷できるよう構成することもできる。

【0178】図26は、本発明の第一の実施例における検索クライアントの動作手順を示すPAD図である。キーボード202からのクライアント起動コマンド等によりクライアント制御プログラム2503が起動されると、本プログラムはユーザから文書検索を指示するコマンドを受け取ってはその処理を行うループに入る(ステップ2601)。このループは、キーボード202からクライアントの停止を指示するコマンドが入力されるまで継続する。

【0179】ステップ2601のループは、ステップ2602からステップ2605までに示す処理を繰り返す。

【0180】ステップ2602では、検索条件入力プログラム2501を実行し、ユーザとの対話により検索条件を入力し、該検索条件を、文書検索サーバ102が解釈可能な文書検索リクエストに変換する。ステップ2603では、前記文書検索リクエストを、ネットワーク105を介して文書検索サーバ102に送信する。ステップ2604では、文書検索サーバ102から前記文書検索リクエストへの返送として検索結果識別子が返されるのを待ち、該識別子を受信する。

【0181】ステップ2605では、前記検索結果識別子を入力として検索結果表示プログラム2502を実行し、ユーザと対話しつつ検索結果データの問い合わせおよび画面表示を行う。

【0182】図27は、前記図26のステップ2602において実行する検索条件入力プログラム2501の詳細な処理手順を示すPAD図である。検索条件入力プログラム2501は、クライアント制御プログラム2503から起動されると、まず検索条件をユーザが対話的に指定するための画面をディスプレイ201に表示する（ステップ2701）。

【0183】次に、ステップ2702において、前記画面上でユーザが指示した検索条件を読み込む。

【0184】次に、ステップ2703において、前記ステップ2702において読み込んだ検索条件を、文書検索サーバ102が直接解釈可能な文書検索リクエストの形に変換する。

【0185】図28は、前記図26のステップ2605において実行する検索結果表示プログラム2502の詳細な処理手順を示すPAD図である。検索結果表示プログラム2502は、クライアント制御プログラム2503から前記検索結果識別子を入力として起動されると、ただちにステップ2801のループに入る。該ループは、ユーザから検索結果表示の終了を指示されるまで、ステップ2802からステップ2815までに示す処理を、繰り返し実行する。

【0186】前記ステップ2801のループ内では、まずステップ2802において、検索結果の表示とユーザからの指示入力のために用いる画面をディスプレイ201に表示する。次に、ステップ2803において、前記画面上でユーザが指定した指示内容を読み込む。

【0187】次に、ステップ2804において、前記ユーザ指示の種別を判定し、その種別に対応した分岐を行う。すなわち、該指示が検出文書数の表示を求めるものであった場合にはステップ2805およびステップ2806に示す処理に分岐し、該指示が検出文書群の文書識別子リストの表示を求めるものであった場合にはステップ2807およびステップ2808に示す処理に分岐し、該指示が文書内容の表示を求めるものであった場合にはステップ2809からステップ2815までに示す処理に分岐する。各分岐先の処理が終了するとステップ2802に戻り、再び前記ループを再開する。

【0188】ここで、ステップ2805では、検出文書数を問い合わせるための検出文書数問い合わせリクエストを生成し、該リクエストを文書検索サーバ102に送信する。次に、ステップ2806では、前記リクエストに対応して文書検索サーバ102から転送されてきた検出文書数を受信し、該数値をディスプレイ201に表示する。

【0189】ステップ2807では、検出文書群の文書

識別子リストを問い合わせるための文書識別子問い合わせリクエストを生成し、該リクエストを文書検索サーバ102に送信する。次に、ステップ2808では、前記リクエストに対応して文書検索サーバ102から転送されてきた文書識別子の集合を受信し、該集合に含まれる文書識別子群をディスプレイ201にリスト表示する。

【0190】ステップ2809では、表示すべき文書を特定する文書識別子を入力する。次に、ステップ2810では、該識別子が識別する文書の解析済み文書データを得るための文書転送リクエストを生成し、該リクエストを文書検索サーバ102に送信する。次に、ステップ2811では、前記リクエストに対応して文書検索サーバ102から転送されてきた解析済み文書データを受信し、該データを解析済み文書データ格納領域217に格納する。

【0191】次に、ステップ2812では、前記解析済み文書データ中において、検索条件中に指定した指定文字列が検出された位置を問い合わせるための文字位置情報問い合わせリクエストを生成し、該リクエストを文書検索サーバ102に送信する。次に、ステップ2813では、前記リクエストに対応して文書検索サーバ102から転送されてきた構造化文字位置情報のリストを受信し、該リストを検索結果データ格納領域1704に格納する。

【0192】次に、ステップ2814では、前記ステップ2811で受信した解析済み文書データおよび前記ステップ2813で受信した構造化文字位置情報リストを参照し、文書検索時における指定文字列検出部分を反転表示するためのデータ加工処理を行う。次に、ステップ2815では、前記反転処理済みの解析済み文書データを、書式化してディスプレイ201上に表示する。

【0193】以上が、本発明の第一の実施例における検索クライアント103および104の動作手順の説明である。

【0194】（第二の実施例）以下、本発明を適用した第二の実施例について、図面を用いて説明する。

【0195】図29は、本実施例における文書登録サブシステム101の構成を示す図である。

【0196】図29に示す文書登録サブシステム101は、そのハードウェア構成に関しては、図2に示す第一の実施例の場合と変わらない。ただし、主メモリ207中には、第一の実施例において保持するプログラム群に加えて、逆順構造インデックス作成プログラム2901を保持する。また、磁気ディスク装置208中には、第一の実施例において確保する領域群に加えて、逆順構造インデックス格納領域2902が確保される。逆順構造インデックス作成プログラム2901は、登録対象文書の解析済み文書データが持つ論理構造に関する情報を、逆順構造インデックス格納領域2902に格納されている逆順構造インデックスに登録して逆順構造インデック

スを更新する。

【0197】本実施例において、文書登録制御プログラム214は、文書構造解析プログラム210、構造インデックス作成プログラム211、逆順構造インデックス作成プログラム2901、構造化全文データ生成プログラム212および文字列インデックス作成プログラム213の起動および実行制御を行うとともに、これらのプログラムによって生成された解析済み文書データ、構造インデックス、逆順構造インデックスおよび文字列インデックスをネットワーク105を介して文書検索サーバ102に転送する。

【0198】なお、本実施例ではフロッピーディスク205に格納された登録対象文書を入力として読み込む構成としたが、光磁気ディスク、追記型光ディスク等、他種の可搬型媒体から読み込む構成をとることもでき、ネットワーク105を介して転送されてくる文書を入力とする構成をとることもできる。さらに、本実施例では生成された解析済み文書データ、構造インデックス、逆順構造インデックスおよび文字列インデックスを文書検索サーバ102に転送するためにネットワーク105を使用するものとしたが、代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。あるいは、文書登録システム101と文書検索サーバ102を1台のコンピュータ上に実装し、データ転送を行わない構成をとることもできる。

【0199】図30は、本発明の第二の実施例における文書登録処理の概略手順を示すPAD図である。本図に示す処理手順は、図3に示す第一の実施例における処理手順とはほぼ同様であるが、図3におけるステップ305の直後にステップ3001が追加されている点と、ステップ308の代わりにステップ3002を実行する点が異なる。

【0200】ここで、ステップ3001では、ステップ304で生成した解析済み文書データを入力として逆順構造インデックス作成プログラム2901を実行する。逆順構造インデックス作成プログラム2901は、まず逆順構造インデックス格納領域2902から現時点での逆順構造インデックスを読み出し、与えられた解析済み文書データの持つ構造情報をこの逆順構造インデックスに登録し、更新された逆順構造インデックスを再び逆順構造インデックス格納領域2902に格納する。

【0201】ステップ3002では、解析済み文書データ格納領域217に格納されたすべての解析済み文書データ、構造インデックス格納領域218に格納された構造インデックス、逆順構造インデックス格納領域2902に格納された逆順構造インデックス、および文字列インデックス格納領域220に格納された文字列インデックスを、ネットワーク105を介して文書検索サーバ102に転送する。

【0202】図31は、図30におけるステップ3001の詳細、すなわち本実施例における逆順構造インデックス作成プログラム2901の処理手順を示すPAD図である。

【0203】逆順構造インデックス作成プログラム2901は、まずステップ3101において、既存の逆順構造インデックスが逆順構造インデックス格納領域2902中に存在しているかどうかを判定する。該領域中に逆順構造インデックスが存在しない場合にはステップ3102に進み、初期状態（空）の逆順構造インデックスを生成する。既存の逆順構造インデックスを検出した場合にはステップ3103に進み、該逆順構造インデックスを読み込む。

【0204】次に、ステップ3104において、登録対象文書の解析済み文書データを読み込む。

【0205】次に、ステップ3105において、前記解析済み文書データの木構造を構成するすべてのノード（要素および文字列データ）を対象として、ステップ3106からステップ3109に示す処理を繰り返す。

【0206】ステップ3106では、解析済み文書データ中で現在着目しているノードについて、逆順構造インデックス中に、該ノードに対応するメタノード（メタ要素またはメタ文字列データ）が存在するかどうかを判定する。対応メタノードが存在しない場合ステップ3107に進み、該ノードに対応するメタノードを生成して逆順構造インデックスに登録し、さらに該登録したメタノードに対して、該メタノードを逆順構造インデックス中で一意に識別する番号である逆順文脈識別子を割り当てる（ステップ3108）。

【0207】ステップ3109では、解析済み文書データ中で現在着目しているノードと、逆順構造インデックス中で該ノードに対応するメタノードを識別する逆順文脈識別子との対応関係を、解析済み文書データに付加して該解析済み文書データを更新する。

【0208】ステップ3105以下の繰り返しを終了するとステップ3110に進み、前記更新された解析済み文書データを出力して解析済み文書データ格納領域217に格納する。次に、ステップ3111では、前記更新された逆順構造インデックスを出力して逆順構造インデックス格納領域2902に格納し、処理を終了する。

【0209】以上に述べたとおり、逆順構造インデックス作成プログラム2901の処理手順は図9に示す構造インデックス作成プログラム211の処理手順とはほぼ対応している。しかし、ステップ3105の繰り返しにおいて解析済み文書の木構造を辿る順序が構造インデックス作成プログラム211の場合とは異なっており、その結果構築される逆順構造インデックスの木構造も構造インデックスの木構造とは異なったものとなる。

【0210】ここで、前記ステップ3105において解析済み文書データの木構造を構成するすべてのノードを

対象とする繰り返しを行う際に、該木構造を辿り、個々のノードを処理していく順序を図 3 2 を用いて説明する。図 3 2 では、楕円形で要素ノード、矩形で文字列ノードを表現し、あるノードが複数の下位ノードを持つ場合には、各下位ノードをその出現順に左から右に並べて表現している。また、各ノードに表示された数字がその処理順序を示している。

【0 2 1 1】図 3 2 に示すとおり、ステップ 3 1 0 5 においてノード群を処理していく順序は、木構造の根に位置するノードから出発し、ある特定のノードとその下位ノード群を処理する際には、まず該ノードの処理を行い、次に該ノードの下位ノード群を、その出現順の逆順に従って処理していく順序となる。

【0 2 1 2】次に、前記ステップ 3 1 0 6 における処理、すなわち解析済み文書データ中で現在着目しているノードについて、逆順構造インデックス中に、該ノードに対応するメタノードが存在するかどうかを判定する処理について、図 3 3 を用いて説明する。図 3 3 は、図の左側に示す解析済み文書データの木構造を構成するノード群と、右側に示す逆順構造インデックスの木構造を構成するノード（メタノード）群との対応関係を示す図である。

【0 2 1 3】ここで、本実施例では、解析済み文書の木構造中におけるあるノードの逆順木構造アドレスと、逆順構造インデックスの木構造中におけるあるメタノードの逆順木構造アドレスとが等しい場合に、該ノードと該メタノードとは対応するものと定義する。ここでいう逆順木構造アドレスとは、木構造の根から出発して上位ノードから下位ノードへと辿り、特定のノードに至るまでの道筋を、該道筋上に存在する各ノードの種別（要素か文字列データか、要素の場合、どの要素型に属するか）と、それらのノードが種別の等しい兄弟ノード群中で後ろから何番目に現れたかを示す番号との組み合わせによって表現するアドレスのことである（通常の木構造アドレスと区別するため、逆順木構造アドレスでは前記番号を負の整数として表記する）。

【0 2 1 4】例えば、図 3 3 に示す解析済み文書データ中のノード群のうち、ノード 3 3 0 1 は上位ノードをもち、兄弟ノード中では最後の“論文”要素ノードであるから、その逆順木構造アドレスは“/論文[-1]”と表記することができる。同様に、ノード 3 3 0 2 はノード 3 3 0 1 の下位ノードであり、兄弟ノード中では最後の“章”要素ノードであるから、その逆順木構造アドレスは“/論文[-1]/章[-1]”と表記することができる。また、ノード 3 3 0 3 はノード 3 3 0 2 の下位ノードであり、兄弟ノード中では後ろから 2 番目の“節”要素ノードであるから、その逆順木構造アドレスは“/論文[-1]/章[-1]/節[-2]”と表記することができる。また、ノード 3 3 0 4 はノード 3 3 0 3 の下位ノードであり、兄弟ノード中では最後の“段落”要素ノードであるから、そ

の逆順木構造アドレスは“/論文[-1]/章[-1]/節[-2]/段落[-1]”と表記することができる。

【0 2 1 5】同様に、図 3 3 右側の構造インデックスの木構造を構成する各メタノードについてその逆順木構造アドレスを求めてみると、メタノード 3 3 0 5 の逆順木構造アドレスは“/論文[-1]”となり、ノード 3 3 0 1 の逆順木構造アドレスと等しくなる。同様に、メタノード 3 3 0 6 の逆順木構造アドレスは“/論文[-1]/章[-1]”となってノード 3 3 0 2 の逆順木構造アドレスと等しく、メタノード 3 3 0 7 の逆順木構造アドレスは“/論文[-1]/章[-1]/節[-2]”となってノード 3 3 0 3 の逆順木構造アドレスと等しくなる。よって、前記ステップ 3 1 0 6 において、ノード 3 3 0 1 はメタノード 3 3 0 5、ノード 3 3 0 2 はメタノード 3 3 0 6、ノード 3 3 0 3 はメタノード 3 3 0 7 とそれぞれ対応するものと判定される。なお、図 3 3 の構造インデックス中にはノード 3 3 0 4 と等しい逆順木構造アドレスを持つメタノードはないので、ノード 3 3 0 4 と対応するメタノードは逆順構造インデックス中に存在しないものと判定され、前記ステップ 3 1 0 7 において新たなメタノードが生成され、該メタノードが構造インデックスに登録されることになる。

【0 2 1 6】前記ステップ 3 1 0 7 において、あるノードに対応する新たなメタノードを登録する際には、該ノードの上位ノードに対応するメタノードの持つ下位メタノード群の先頭に、該ノードに対応する種別のメタノードを追加する。図 3 3 のノード 3 3 0 4 に対応するメタノードを登録する場合には、ノード 3 3 0 4 の上位ノードであるノード 3 3 0 3 に対応するメタノード 3 3 0 7 の下位に、要素型“段落”のメタノードが追加され、該メタノードは兄弟メタノード群の先頭に置かれることになる。

【0 2 1 7】次に、複数の解析済み文書データを順次重ね合わせることで逆順構造インデックスを生成していく過程について、図 3 4 を用いて説明する。図 3 4 において、3 4 0 1、3 4 0 3 および 3 4 0 5 は、それぞれ登録対象文書の解析済み文書データを表わしている。これらの解析済み文書データの構造を既存の逆順構造インデックス上に順次重ね合わせることで、逆順構造インデックスが形成されていく。

【0 2 1 8】まず最初に文書 1 の解析済み文書データ 3 4 0 1 が入力されると、最初の段階では逆順構造インデックスは初期状態（空）であるため、該解析済みデータと等価な木構造が生成されてそのまま逆順構造インデックスに登録され、逆順構造インデックスは 3 4 0 2 に示す状態となる。新たに生成されたメタ要素には -E 1 から -E 5 までの文脈識別子、新たに生成されたメタ文字列データには -C 1 から -C 3 までの文脈識別子が割り当てられる。

【0 2 1 9】次に文書 2 の解析済み文書データ 3 4 0 3

が入力されると、既存の逆順構造インデックス（3 4 0 2）と構造が重複する部分については何も行わず、3 4 0 2 上に対応する部分がなかった部分構造（図中の斜線部分）だけが新たに登録される。新たに生成されたメタ要素には文脈識別子-E 6 および-E 7、新たに生成されたメタ文字列データには文脈識別子-C 4 が割り当てられる。

【0 2 2 0】次に文書 3 の解析済み文書データ 3 4 0 5 が入力されると、既存の逆順構造インデックス（3 4 0 4）と構造が重複する部分については何も行わず、3 4 0 4 上に対応する部分がなかった部分構造（図中の斜線部分）だけが新たに登録される。新たに生成されたメタ要素には文脈識別子-E 8、-E 9 および-E 1 0、新たに生成されたメタ文字列データには文脈識別子-C 5 および-C 6 が割り当てられる。このようにして、3 個の文書が登録された段階で、逆順構造インデックスは 3 4 0 6 に示す状態となる。

【0 2 2 1】図 3 5 は、図 3 0 におけるステップ 3 0 6 の詳細、すなわち本実施例における構造化全文データ生成プログラム 2 1 2 の処理手順を示す P A D 図である。図 3 5 に示すとおり、本実施例における構造化全文データ生成プログラム 2 1 2 の処理手順は、図 1 3 に示す前記第一の実施例の場合の処理手順とほぼ同一である。ただし、本実施例の場合、図 1 3 におけるステップ 1 3 0 5 の代わりにステップ 3 5 0 1 を行う点が異なっている。

【0 2 2 2】ここで、ステップ 3 5 0 1 では、解析済み文書データから、現在着目している文字列データノードに対応する文脈識別子および逆順文脈識別子を求め、該文脈識別子および該逆順文脈識別子を構造化全文データ格納領域 2 1 9 に出力する。

【0 2 2 3】図 3 6 に、本実施例において構造化全文データ生成プログラム 2 1 2 が出力する構造化全文データのファイル形式を示す。図 3 6 は、図 5 に示した S G M L 文書を入力として構造化全文データを生成した場合について例示している。図 3 6 に示すとおり、本実施例における構造化全文データのデータファイルは、先頭に文書識別子を記述し、その後ろに、文脈識別子、逆順文脈識別子およびそれらに対応する内容文字列からなる三つ組みを、文書中に存在する文字列データの数だけ繰り返した構造をとる。

【0 2 2 4】本実施例における文字列インデックスは、前記第一の実施例の場合と同じく、図 1 5 に示す処理手順に従って作成される。図 3 7 に、本実施例における文字列インデックスのデータ構造を示す。図 3 7 は、文字列インデックス作成プログラム 2 1 3 を用いて図 3 6 に示す構造化全文データを処理し、文字列インデックスに該構造化全文データに含まれる部分文字列群の登録を終えた段階において、該文字列インデックスのデータ構造の一部（内容文字列“変換処理の一例”に関連する部

分）を図示したものである。

【0 2 2 5】図 3 7 に示すとおり、本実施例における文字列インデックスでは、個々の構造化文字位置情報中に、前記第一の実施例の場合に保持する情報に加えて、逆順文脈識別子をも保持する。ただし、図 3 7 では、前記第一の実施例における図 1 6 と同様、前記内容文字列の末尾“例”に対応する文字ノードおよび構造化文字位置情報は省略している。また、前記内容文字列の直前に位置する文字の文字位置を“X”として、相対的に文字位置を表現している。

【0 2 2 6】以上が、本実施例における文書登録サブシステム 1 0 1 の説明である。

【0 2 2 7】以下、本発明の第二の実施例における文書検索サーバ、すなわち図 1 の 1 0 2 について説明する。

【0 2 2 8】図 3 8 は、本実施例における文書検索サーバ 1 0 2 の構成を示す図である。図 3 8 に示すとおり、本実施例における文書検索サーバ 1 0 2 は、前記第一の実施例の場合の構成要素群に加えて、磁気ディスク装置 2 0 8 中に、逆順構造インデックス格納領域 2 9 0 2 を保持する。

【0 2 2 9】なお、本実施例の場合も、文書登録サブシステム 1 0 1 および検索クライアントとの間でデータを転送するためにネットワーク 1 0 5 を使用する代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。また、文書登録サブシステム 1 0 1 と文書検索サーバ 1 0 2 を 1 台のコンピュータ上に実装し、これらの間でデータ転送を行わない構成をとることもできる。また、1 個以上の検索クライアントを文書検索サーバ 1 0 2 と同一のコンピュータ上で実行し、これらの間でデータ転送を行わない構成をとることもできる。

【0 2 3 0】図 3 9 は、本発明の第二の実施例における文書検索処理の概略手順を示す P A D 図である。図 3 9 に示すとおり、本実施例における文書検索処理の手順は、図 1 8 に示す前記第一の実施例の場合とほぼ同一であるが、前記第一の実施例におけるステップ 1 8 0 5 の代わりにステップ 3 9 0 1 を実行する点が異なっている。ステップ 3 9 0 1 では、文書登録サブシステム 1 0 1 から、新規に登録された文書群の内容を反映して更新された構造インデックス、逆順構造インデックスおよび文字列インデックスを受信し、これらをそれぞれ構造インデックス格納領域 2 1 8、逆順構造インデックス格納領域 2 9 0 2 および文字列インデックス格納領域 2 2 0 に格納する。

【0 2 3 1】図 4 0 は、図 3 9 におけるステップ 1 8 0 6 の詳細、すなわち本実施例における検索条件解析プログラム 1 7 0 1 の処理手順を示す P A D 図である。

【0 2 3 2】本実施例の場合、検索条件解析プログラム 1 7 0 1 は、文書検索リクエスト中で指定されている検索条件を入力として起動されると、まずステップ 4 0 0

1において、該検索条件中に含まれている構造条件について判定する。ここで、該検索条件中に正順の（すなわち前記第一の実施例におけるものと同様な）構造条件が含まれている場合には、ステップ1902およびステップ1903からなる処理に分岐する。ここで、ステップ1902および1903での処理は、前記第一の実施例の場合と同一である。該検索条件中に逆順の構造条件が含まれている場合には、ステップ4002およびステップ4003からなる処理に分岐する。該検索条件中に構造条件が含まれていなかった場合には何もせずにステップ1904に進む。

【0233】ステップ4002では、逆順構造インデックス格納領域2902から逆順構造インデックスを読み込む。次に、ステップ4003では、該逆順構造インデックスを参照して、前記構造条件を満たす構造内に含まれるすべての文字列データの逆順文脈識別子の集合を求める。以下、該集合を逆順文脈識別子集合と呼ぶ。

【0234】ステップ1904およびそこから分岐するステップ1905およびステップ1906における処理は、前記第一の実施例の場合と同一であり、これらの処理を終了した後、ステップ4004に進む。

【0235】ステップ4004では、ステップ1903で求めた文脈識別子集合、前記ステップ4003で求めた逆順文脈識別子集合、前記検索条件中に含まれていた指定文字列、および前記ステップ1905またはステップ1096で生成した部分文字列リストから構成される展開済み部分文字列データを生成して処理を終了する。

【0236】ここで、図41は、本実施例での検索条件解析プログラム1701の処理過程における、展開済み解析条件データの生成例を示す図である。

【0237】図41において、4101は、文書検索リクエスト中で指定された検索条件の一例である。検索条件4101は、構造条件指定“章/段落[-1]”と、文字列条件指定“ガード”とから構成されている。前記検索条件は、“章”要素の直接の下位にある最後の“段落”要素内に、文字列“ガード”が出現するケースを検索すべきことを指定している。

【0238】ここで、前記検索条件中で指定されている構造条件は、構造を後ろ側から辿って条件を指定する逆順の構造条件であるから、逆順構造インデックスの内容が4102に示すとおりであったとすると、前記ステップ4003においてこの構造インデックスを参照することにより、前記構造条件指定を満たす“段落”要素の逆順文脈識別子は-E3および-E12であることがわかる。従って、これらの段落の下位にある文字列データ、すなわち逆順文脈識別子が-C1または-C7である文字列データ内に、文字列“ガード”が出現するケースを検索すればよいことがわかる。ただし、検索に用いる文字列インデックスには、長さ2の部分文字列についてのみその出現位置が登録されているので、3文字から

なる前記指定文字列を直接検索することはできない。ここで、ステップ1905において、前記指定文字列を分解して長さ2の部分文字列からなるリストを生成する。前記のとおり指定文字列が“ガード”だった場合、抽出される部分文字列は“ガー”および“ード”となる。

【0239】この結果、前記ステップ4004において、4103に示す展開済み検索条件データ、すなわち文脈識別子集合が空集合、逆順文脈識別子集合が{-C1, -C7}、指定文字列が“ガード”、部分文字列リストが{“ガー”, “ード”}であるデータが生成される。

【0240】図42は、図39におけるステップ1807の詳細、すなわち本実施例における文字列インデックス検索プログラム1702の処理手順を示すPAD図である。

【0241】文字列インデックス検索プログラム1702は、前記検索条件解析プログラム1701が生成した展開済み検索条件データを入力として起動される。図42に示すとおり、本プログラムの処理手順は図21に示す第一の実施例の場合の処理手順とはほぼ同様であるが、前記図21におけるステップ2104、ステップ2105およびステップ2107に代わって、それぞれステップ4201、ステップ4202およびステップ4203が実行される。

【0242】ステップ4201では、文字列インデックス中で前記展開済み検索条件データ中における指定文字列を検索し、該文字列に対応している構造化文字位置の集合を求め、次に該集合の中から、前記展開済み検索条件データ中の文脈識別子集合に含まれているいずれかの文脈識別子、または前記展開済み検索条件データ中の逆順文脈識別子集合に含まれているいずれかの逆順文脈識別子を持つ構造化文字位置情報群のみを抽出し、該抽出した構造化文字位置情報群からなるヒット位置集合を生成する。

【0243】ステップ4202では、文字列インデックス中で前記指定文字列を検索し、該文字列の末尾に対応する文字ノードより前方に存在するすべての構造化文字位置情報の集合を求め、次に該集合の中から、前記展開済み検索条件データ中の文脈識別子集合に含まれているいずれかの文脈識別子、または前記展開済み検索条件データ中の逆順文脈識別子集合に含まれているいずれかの逆順文脈識別子を持つ構造化文字位置情報群のみを抽出し、該抽出した構造化文字位置情報群からなるヒット位置集合を生成する。

【0244】ステップ4203では、前記展開済み検索条件データ中における部分文字列リスト中に含まれている部分文字列群のうち、ステップ2106の繰り返し中で現在着目している部分文字列を文字列インデックス中で検索し、該文字列に対応している構造化文字位置情報の集合を求め、次に該集合の中から、前記展開済み検索

条件データ中の文脈識別子集合に含まれているいずれかの文脈識別子、または前記展開済み検索条件データ中の逆順文脈識別子集合に含まれているいずれかの逆順文脈識別子を持つ構造化文字位置情報群のみを抽出し、該抽出した構造化文字位置情報群を前記部分文字列に対応付けて記憶する。

【0245】図42におけるステップ2108、すなわち文字列インデックス検索プログラム1702の処理過程における接続判定処理は、図22に示す前記第一の実施例の場合と同様となる。ただし、検索条件中の構造条件が逆順の構造条件だった場合には、接続判定の際、文脈識別子の一致ではなく逆順文脈識別子の一致を判定することになる。

【0246】以上説明したように、本実施例に示した構成によれば、前記第一の実施例において検索が可能となる各種構造条件に加えて、「論文中で最後の章の中で、ある特定の文字列を検索する」、あるいは「後ろから2番目の参考文献の中で、ある特定の文字列を検索する」のように、文書の論理構造を逆順に（後ろから）辿った構造条件を指定した検索も可能となる。

【0247】以上が、本発明の第二の実施例の説明である。

【0248】（第三の実施例）以下、本発明を適用した第三の実施例について、図面を用いて説明する。

【0249】本実施例は、システムの構成および各プログラムの処理手順のいずれについても前記第一の実施例と同一であるが、文書木構造中のノードと構造インデックス中のメタノードとの対応のさせ方が異なり、その結果、同一の文書群を入力とした場合でも構造インデックスの構造および文脈インデックスの割り当てが前記第一の実施例とは異なってくる。

【0250】ここで、本実施例における、文書木構造中のノードと構造インデックス中のメタノードとの対応関係を、図43を用いて説明する。図43は、図の左側に示す解析済み文書データの木構造を構成するノード群と、右側に示す構造インデックスの木構造を構成するノード（メタノード）群との対応関係を示している。

【0251】本実施例においても、解析済み文書の木構造中におけるあるノードの木構造アドレスと、構造インデックスの木構造中におけるあるメタノードの木構造アドレスとが等しい場合に、該ノードと該メタノードとは対応するものと定義する。ただし、本実施例では、前記第一の実施例とは異なり、共通の上位ノードを持つ同種の兄弟ノード間での出現順位を考える際に、先頭のノードと2番目のノードとの区別は行わぬが、2番目のノードとそれ以降に現れたノードとの区別は行わない。すなわち、木構造アドレス中で出現順位を表わす番号は常に[1]または[2]のいずれかをとり、[3]以上の値をとることはない。

【0252】例えば、図43に示す解析済み文書データ

中のノード群のうち、ノード4301は上位ノードをもたず、兄弟ノード中では最初の“論文”要素ノードであるから、その木構造アドレスは“/論文[1]”と表記することができる。同様に、ノード4302はノード4301の下位ノードであり、兄弟ノード中では最初の“章”要素ノードであるから、その木構造アドレスは“/論文[1]/章[1]”となる。これに対して、ノード4303はノード4302の下位ノードであり、兄弟ノード中では4番目の“節”要素ノードであるが、前記規則から、その木構造アドレスは“/論文[1]/章[1]/節[2]”となる。また、ノード4304はノード4303の下位ノードであり、兄弟ノード中では2番目の“段落”要素ノードであるから、その木構造アドレスは“/論文[1]/章[1]/節[2]/段落[2]”となる。

【0253】同様にして、図43右側の構造インデックスの木構造を構成する各メタノードについてその木構造アドレスを求めてみると、メタノード4305の木構造アドレスは“/論文[1]”となり、ノード4301の木構造アドレスと等しくなる。同様に、メタノード4306の木構造アドレスは“/論文[1]/章[1]”となってノード4302の木構造アドレスと等しく、メタノード4307の木構造アドレスは“/論文[1]/章[1]/節[2]”となってノード4303の木構造アドレスと等しくなる。よって、ノード4301はメタノード4305、ノード4302はメタノード4306、ノード4303はメタノード4307とそれぞれ対応するものと判定される。なお、図43の構造インデックス中にはノード4304と等しい木構造アドレスを持つメタノードはないので、ノード4304と対応するメタノードは構造インデックス中に存在しないものとみなされる。

【0254】以上説明したように、上記規則の結果、本実施例において形成される構造インデックスでは、同一のメタノードの下位に、同種のメタノードが3個以上付加されることはなく、文書木構造中で同種のノードが3個以上連続している場合、2番目以降のノードには同一の文脈識別子が割り当てられる。よって、本実施例による文書検索方法では、構造条件中で任意の出現順位を指定することはできず、同種要素中で先頭の要素か、2番目以降の要素かを区別できるだけとなる。その代わり、本実施例では構造インデックスのデータ構造が前記第一の実施例の場合と比べて簡略になり、構造インデックス格納領域218として必要となる容量を削減することができる。

【0255】なお、本実施例におけるノードとメタノードとの対応付けを採用した場合においても、前記第二の実施例同様に正順・逆順二つの構造インデックスを用意することにより、出現順を後ろから辿った構造指定をも可能とすることができる。

【0256】以上が、本発明の第三の実施例の説明である。

47

【0257】（第四の実施例）以下、本発明を適用した第四の実施例について、図面を用いて説明する。

【0258】図44は、本実施例における文書登録サブシステム101の構成を示す図である。

【0259】図44に示す文書登録サブシステム101は、そのハードウェア構成および磁気ディスク208中の格納領域の構成に関しては、図2に示す第一の実施例の場合と変わらない。ただし、主メモリ207中には、第一の実施例において保持するプログラム群に加えて、文書構造正規化プログラム4401を保持する。

【0260】本実施例において、文書登録制御プログラム214は、文書構造解析プログラム210、文書構造正規化プログラム4401、構造インデックス作成プログラム211、構造化全文データ生成プログラム212および文字列インデックス作成プログラム213の起動および実行制御を行うとともに、これらのプログラムによって生成された解析済み文書データ、構造インデックスおよび文字列インデックスをネットワーク105を介して文書検索サーバ102に転送する。

【0261】なお、本実施例ではフロッピーディスク205に格納された登録対象文書を入力として読み込む構成としたが、光磁気ディスク、追記型光ディスク等、他種の可搬型媒体から読み込む構成をとることもでき、ネットワーク105を介して転送されてくる文書を入力とする構成をとることもできる。さらに、本実施例では生成された解析済み文書データ、構造インデックスおよび文字列インデックスを文書検索サーバ102に転送するためにネットワーク105を使用するものとしたが、代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。あるいは、文書登録サブシステム101と文書検索サーバ102を1台のコンピュータ上に実装し、データ転送を行わない構成をとることもできる。

【0262】図45は、本発明の第四の実施例における文書登録処理の概略手順を示すPAD図である。本図に示す処理手順は、図3に示す第一の実施例における処理手順とはほぼ同様であるが、ステップ304の直後にステップ4501が追加されている点が異なる。

【0263】ここで、ステップ4501では、ステップ304で生成した解析済み文書データを入力として文書構造正規化プログラム4401を実行する。文書構造正規化プログラム4401は、解析済み文書データの中から、検索対象としては不適切な構造および内容文字列を抽出してこれらを削除する。

【0264】図46に、文書構造正規化プログラム4401の処理手順を示す。本プログラムは、起動されるとまずステップ4601において、正規化パラメタの指定の有無を確認する。正規化パラメタが指定されている場合にはステップ4602からステップ4608までに示す処理を実行し、正規化パラメタが指定されていない場

48

合には何もせずに処理を終了する。

【0265】ここで、正規化パラメタとは、接続対象要素および削除対象要素の要素型名を指定するパラメタのことである。ここで言う接続対象要素とは、例えば文章の一部を強調表示するために用いられる非構造的な要素であって、検索時にはその要素の境界をまたいで文字列を検出する必要があるもののことである。また、削除対象要素とは、その内部に本来の文書内容とは異なる種類のデータを保持しており、検索時にはその内容を無視して文字列の検出を行うべきもののことである。削除対象要素の例としては、例えばテキスト中に参照先の文献へのリンクデータを埋め込むために用いられる要素などがある。

【0266】ステップ4602では、指定されている正規化パラメタを読み込む。ここで、正規化パラメタは、ユーザがキーボード202から入力するか、または予め特定のファイルに書き込んでおくなどして指定する。接続対象要素および削除対象要素の要素型名は、それぞれ複数個指定してもよく、あるいは省略してもよい。次に、ステップ4603では、解析済み文書データを解析済み文書データ格納領域217から読み込む。

【0267】次に、ステップ4604では、解析済み文書データの本構造を順次たどりつつ、すべての要素ノードについて、ステップ4605からステップ4607までに示す処理を繰り返す。そして、すべての要素ノードについて該処理を終了した後、ステップ4608に進む。

【0268】ステップ4605では、現在着目している要素ノードについて、正規化パラメタが何を指定しているかを判定する。該着目要素の要素型名が接続対象として指定されている場合にはステップ4606に進み、該要素ノードを削除するとともに、該要素の内部に含まれるすべての文字列データを、該要素の前後にある文字列データに接続する。また、該着目要素の要素型名が削除対象として指定されている場合にはステップ4607に進み、該要素ノードおよびその下位にあるすべてのノードを削除する。

【0269】前記のごとくステップ4605以下の処理を行った後、ステップ4608に進み、正規化対象要素群を処理して更新された解析済み文書データを、解析済み文書データ格納領域217に再び格納する。

【0270】図47は、正規化処理の具体的な例を示す図である。

【0271】図47において、4701は正規化パラメタの一例を示しており、ここでは接続対象要素の要素型名として“bold”と“italic”の二つ、削除対象要素の要素型名として“link”と“index”の二つが指定されている。この場合、解析済み文書データ中に4702に示すような構造を持つ部分があると、要素“bold”に対して前記ステップ4606に示す接続処理が行われ、そ

の結果は4703に示す構造となる。また、解析済み文書データ中に4704に示すような構造を持つ部分があると、要素“link”に対して前記ステップ4607に示す削除処理が行われ、その結果は4705に示す構造となる。

【0272】以上説明したように、本実施例では、解析済み文書データに対して前記正規化処理を行った後に構造インデックスへの登録以下の処理が行われるので、登録対象文書中に含まれていた非構造的な要素に妨げられることなくテキストの検索を行うことができる。

【0273】なお、本実施例における正規化処理を採用した場合においても、前記第二の実施例同様に正順・逆順二つの構造インデックスを用意することにより、出現順を後ろから辿った構造指定も可能とすることができる。

【0274】以上が、本発明の第四の実施例の説明である。

【0275】（第五の実施例）本発明を利用した第五の実施例について、図面を用いて説明する。

【0276】図48は、本実施例における文書登録サブシステム101の構成を示す図である。

【0277】本システム構成は、そのハードウェア構成においては、図2に示す第一の実施例の文書登録サブシステム101と変わらない。ただし、主メモリ207に格納される文書登録プログラムの1つである、構造インデックス作成プログラム211を、メタ構造インデックス作成プログラム4801に変更している。さらに、磁気ディスク装置208には、構造インデックス格納領域218の代わりにメタ構造インデックス格納領域4802を作成し、種別定義テーブル格納領域4803を追加する。

【0278】メタ構造インデックス作成プログラム4801は、文書構造解析プログラム210の出力結果である解析済み文書データ217を入力として、様々な構造を持つ登録文書の全ての文書構造を一括管理するメタ構造インデックスを出力する。

【0279】図49にメタ構造インデックスの例を示す。本図は、“論文”という種別の最上位構造を持つ構造インデックス1(4901)と“報告書”という種別の最上位構造を持つ構造インデックス2(4902)をルートメタノード4903によって1つの木構造にまとめたものである。この木構造をメタ構造インデックス4904と呼ぶ。

【0280】つまり、構造インデックスは、最上位構造の種別が一致する登録文書の構造との重ね合わせにより作成していくため、最上位構造の種別が同じ登録文書群毎に作成することになる。メタ構造インデックスとは、これらの異なる最上位構造を持つ全ての構造インデックスの最上位構造を、1つのルートメタノードに接続することで、1つのインデックスにまとめたものである。

【0281】ルートメタノードとは、複数の構造インデックスの最上位構造をまとめる仮の上位構造である。つま

り、ルートメタノードから、複数の構造インデックスを辿るために存在しており、ルートメタノードに対応する構造が登録文書に存在するわけではない。

【0282】ルートメタノードは、構造インデックスの要素メタノードと同じく、下位要素である複数の構造インデックスの最上位構造の数および各構造インデックスへのリンクの情報を持つ。

【0283】上記のメタ構造インデックスを格納するのが、メタ構造インデックス格納領域4802である。

10 【0284】次に、種別定義テーブル4803について説明する。

【0285】種別定義テーブルは、構造化文書の各構造に付けられた要素型名とその意味を表わす種別との対応を定義したテーブルである。ユーザがキーボード202からテキストエディタを利用して作成するなどの方法で、あらかじめ磁気ディスク装置208に格納しておく。

【0286】図50は種別定義テーブル4803の内容を示す図である。種別とは、例えば“論文”、“PAPER”のように、名称が違っていても同じ意味を持つ要素型名を代表する名称である。種別定義テーブルは、複数の要素型名に付けられた種別を管理しており、本テーブルにおいて同じ種別が定義されている要素型名を持つ構造は、同じ種別であると判断する。

【0287】本図に示したように、種別定義テーブルには、種別5001と要素型名数5002と要素型名5003の3つの情報を格納する。種別5001が、複数の要素型名に共通に付けられる種別である。要素型名数5002は、各種別に対応する要素型名の数である。さらに、要素型名5003には、要素型名数に記載された数の要素型名を列挙する。

【0288】本テーブルを参照することにより、要素型名から種別の情報を得ることができる。また、逆に、種別から要素型名を得ることができる。さらに、本テーブルに記載されない要素型名は、その要素型名をそのまま種別とする。

【0289】本実施例において、種別定義テーブル4803に記載される種別と要素型名は、1対多の対応を満たす。つまり、“書誌”という要素型名は一意に1つの種別を持つ。これは、種別定義テーブルをメタ構造インデックスごとに作成しており、メタ構造インデックスの各メタノードは、種別および出現位置により区別されるためである。要素型名によって一意に種別が決まらなければ、構造インデックスの作成において、要素型名から得られる複数の種別のうちいずれを用いるかが決まらないためである。

【0290】図51は、本実施例における文書登録処理の概略手順を示すPAD図である。本図は、図3に示す第一の実施例の登録処理の概略手順とほぼ同じであるが、図3におけるステップ305の代わりにステップ5

101を実行する点と、ステップ308の代わりにステップ5102を実行する点異なる。

【0291】ステップ5101では、メタ構造インデクス作成プログラム4801を呼び出す。メタ構造インデクス作成プログラム4801では、登録済みのメタ構造インデクスをメタ構造インデクス格納領域4802から読み出し、ステップ304で得られた解析済み文書データの持つ構造情報を、このメタ構造インデクスに登録し、更新されたメタ構造インデクスをメタ構造インデクス格納領域4802に格納する。

【0292】ステップ5102では、登録文書全ての解析済み文書データ217および更新されたメタ構造インデクス4804、文字列インデクス220を文書検索サーバ102に転送する。

【0293】図52は、ステップ5101でメタ構造インデクスを作成する処理の詳細内容を示すPAD図である。本処理は、図9に示した第一の実施例における構造インデクス作成プログラムの処理とほぼ同じ内容であるが、登録先をメタ構造インデクスに変更していることから、以下の点で図9と異なる。

【0294】まず、ステップ904の解析済み文書データ読み込み処理を実行する。

【0295】次に、ステップ901の代わりにステップ5201を実行する。ステップ901は、構造インデクスそのものの有無を判定しているが、ステップ5201では、構造インデクスは1つのメタ構造インデクスの一部であり、最上位構造の種別毎に作成することから、メタ構造インデクス中に、登録文書の最上位構造と一致する構造インデクスが存在するか否かをチェックする処理に変更する。

【0296】ステップ904をステップ5201の前に実行するのは、本処理において、登録文書の最上位構造の情報が必要なためである。ここで、最上位構造の種別が一致する構造インデクスが存在しない場合は、新しく初期構造インデクスを生成するステップ902の処理を実行し、存在する場合は、該構造インデクスを読み込むステップ903の処理を実行する。また、本ステップにおいて、種別の比較をする前に、種別定義テーブル4803を参照し、登録文書の最上位構造の要素型名を種別に変換してから比較を行う。

【0297】さらに、ステップ906の代わりにステップ5202を実行する。ステップ5202では、種別定義テーブル4803を参照して、解析済み文書データの識別子名を種別に変換してから、図11を用いて前述したステップ906の処理内容と同じ手順で、構造インデクス中の対応するメタノードの有無のチェックを行う。

【0298】さらに、ステップ908の代わりにステップ5203を実行する。ステップ5203では、文脈識別子を割り当てる際に、メタ構造インデクス全体で一意的にメタノードを識別できる文脈識別子を与える。したが

って、構造化全文データ作成プログラム212において、各構造のテキスト列に与えられる文脈識別子は、メタ構造インデクス中のメタノードを一意的に決定するものである。ステップ908の処理に加えて、構造インデクスの識別情報を文脈識別子に加えるなどの方法により、実現可能である。

【0299】さらに、ステップ905による繰り返し処理の後に、ステップ5204を実行する。

【0300】ステップ5204では、ステップ902で構造インデクスを新規に作成した場合に、メタ構造インデクスのルートメタノードに新規に作成した構造インデクスの最上位のメタノードを接続し、メタ構造インデクスに新規に作成した構造インデクスを組み込む処理を行う。

【0301】さらに、ステップ911の代わりにステップ5205を実行する。ステップ5205では、作成したメタ構造インデクスをメタ構造インデクス格納領域4802に格納する。

【0302】その他のステップの処理は、図9を用いて前述した内容と同じである。

【0303】図53、図54は、上記のステップ5101で作成するメタ構造インデクスの例である。図53は、最上位構造の種別が一致する構造インデクスが存在する場合の例を示している。また、図54は、最上位構造の種別が一致する構造インデクスが存在しない場合の例を示している。

【0304】図53では、まずメタ構造インデクス5301に存在する構造インデクス5302と登録文書の構造解析結果の木構造5303との比較を行う。この例では登録文書の最上位構造である“文書”と一致する種別の最上位構造を持つ構造インデクス5302が存在するため、この構造インデクス5302に対して、登録文書の木構造5303の重ね合わせを行う。この場合、“日付”のノード5304が構造インデクス5302に存在しないため、構造インデクス5302に、“日付”のノードを追加して、更新された構造インデクス5305を作成する。構造インデクスの更新に伴い、メタ構造インデクス5301も更新される(5306)。

【0305】図54では、まずメタ構造インデクス5401に存在する構造インデクス5402と登録文書の構造解析結果の木構造5403との比較を行う。この例ではメタ構造インデクス中の構造インデクスの最上位構造は、“論文”(5404)しかなく、登録文書の最上位構造である“報告書”(5405)と一致する最上位構造を持つ構造インデクスは存在しない。このため、登録文書の木構造と同じ構造を持つ、構造インデクス5406を新規に作成する。さらに、作成した構造インデクス5406を、ルートメタノード5407に接続することで、メタ構造インデクスに構造インデクス5406を追加する。構造インデクスの追加に伴い、メタ構造インデクス

53

5401も更新される(5408)。

【0306】上記のように、最上位構造が一致する構造インデックスが存在する場合は、この構造インデックスに対する重ね合わせを行い、存在しない場合は、新たに構造インデックスを作成した上で、ルートメタノードに接続することで、メタ構造インデックスを更新する。

【0307】本実施例の文書登録サブシステム101が、第一の実施例の文書登録サブシステムと異なる点は以上であり、その他の構成および処理内容は全て同じである。

【0308】次に、本発明の第五の実施例における文書検索サーバ、すなわち図1の102について説明する。

【0309】図55は、本実施例における文書検索サーバ102の構成を示す図である。

【0310】本システム構成は、そのハードウェア構成においては、図17に示す第一の実施例の文書検索サーバ102の構成と変わらない。

【0311】ただし、主メモリ207に格納される文書検索処理プログラムのうち検索条件解析プログラム1701をメタ構造インデックス対応検索条件解析プログラム5501に変更し、さらに構造インデックス格納領域218の代わりにメタ構造インデックス格納領域4802を作成し、磁気ディスク装置208に種別定義テーブル格納領域4803を追加する点異なる。

【0312】メタ構造インデックス対応検索条件解析プログラム5501は、検索クライアント103および104から受信した検索リクエスト中に含まれる検索条件式を解析し、文字列インデックス検索プログラム1702によって直接検索可能な条件指定に翻訳する。ここで、検索条件解析プログラム1701が、構造インデックスを利用して、検索条件式を解析するのと違い、メタ構造インデックス対応検索条件解析プログラム5501では、メタ構造インデックスおよび種別定義テーブルを利用して検索条件式を解析する。

【0313】また、メタ構造インデックス格納領域4802には、本実施例において前述した、文書登録サブシステム101で作成したメタ構造インデックスを格納する。種別定義テーブル4803は、ユーザによって文書登録サブシステム101に登録された種別定義テーブルと同じ内容である。

【0314】図56は、本実施例における検索サーバの処理の概略手順を示すPAD図である。本図は、図18に示す第一の実施例の検索サーバの処理の概略手順とは同じである。ただし、図18におけるステップ1805の代わりにステップ5601を実行する点と、ステップ1806の代わりにステップ5602を実行する点異なる。

【0315】ステップ5601では、文書登録サブシステム101から、メタ構造インデックスおよび文字列インデックスを受信し、これらをそれぞれメタ構造インデックス

54

格納領域4802および文字列インデックス格納領域220に格納する。これらのメタ構造インデックスおよび文字列インデックスは、ステップ1804で追加登録した、新規に登録された文書群の内容を反映して更新したものである。

【0316】ステップ5602では、メタ構造インデックス対応検索条件解析プログラム5501を実行し、文書検索リクエスト中で指定されている検索条件を解析し、該検索条件を、文字列インデックス検索プログラム1702によって直接処理可能な条件指定(以下、これを展開済み検索条件データと呼ぶ)に変換する。

【0317】その他のステップの処理は、第一の実施例において、図18を用いて前述した内容と同じである。

【0318】図57は、図56におけるステップ5602の詳細、すなわち本実施例におけるメタ構造インデックス対応検索条件解析プログラム5501の処理手順を示すPAD図である。本図は、図19に示す第一の実施例の検索条件解析プログラム1701の処理手順を示すPAD図とはほぼ同じである。図19におけるステップ1902の代わりにステップ5701を実行する点と、ステップ1903の代わりにステップ5702を実行する点異なる。

【0319】ステップ5701では、メタ構造インデックス格納領域4802からメタ構造インデックスを読み込む。

【0320】次に、ステップ5702では、メタ構造インデックスを参照して、前記構造条件を満たす構造内に含まれるすべての文字列データの文脈識別子集合を求める。ここで検索条件で指定された構造条件が、構造の種別により指定される場合は、そのまま、メタ構造インデックスを辿ることで、構造条件を満たす構造内に含まれる文字列データの文脈識別子を得ることができる。要素型名で指定される場合は、種別定義テーブル4803を参照して種別に変換してから、メタ構造インデックスを辿って、構造条件を満たす構造内に含まれる文字列データの文脈識別子を得る。

【0321】メタ構造インデックスの最上位構造は、各文書の最上位構造を接続したルートメタノードであるため、対応する構造条件は存在しない。

【0322】その他のステップは、第一の実施例において、図19を用いて前述した内容と同じである。

【0323】図58は、メタ構造インデックス対応検索条件解析プログラム5501の処理過程における、展開済み解析条件データの生成例を示す図である。

【0324】図58において、5801は、文書検索リクエスト中で指定された検索条件の一例である。検索条件5801は、構造条件指定“論文/書誌/タイトル”と、文字列条件指定“ガード”とから構成されている。前記検索条件は、“論文”要素の直接の下位にある“書誌”要素の直接の下位にある“タイトル”要素内に、文

字列“ガード”が出現するケースを検索すべきことを指定している。

【0325】ここでは、構造条件に種別を指定した場合について説明する。要素型名を指定した場合は、種別定義テーブル4803を参照して、種別に変換した後に以下の処理を行なう。

【0326】さらに、構造条件に要素型名、種別を混在して指定するために、種別の場合は構造条件の前に“Type:”などの識別情報を付加する。このため、検索クライアント103、104において、第一の実施例で図27を用いて説明したステップ2702およびステップ2703で、ユーザ指示および文書検索リクエストに上記の識別情報を追加する。ただし、本設定を行なっても、メタ構造インデックスには種別の情報しか持たないため、要素型名による構造条件でメタ構造インデックスから構造条件を満たす構造内に含まれる文字列データの文脈識別子が得られるわけではない。

【0327】ここで、メタ構造インデックスの内容が5802に示すとおりであったとすると、前記ステップ5702において、この構造インデックスを参照することにより、前記構造条件指定を満たす“タイトル”要素の文脈識別子はE3であることがわかる。従って、この段落の下位にある文字列データ、すなわち文脈識別子がC1である文字列データ内に、文字列“ガード”が出現するケースを検索すればよいことがわかる。ただし、検索に用いる文字列インデックスには、長さ2の部分文字列についてのみ、その出現位置が登録されているので、3文字からなる前記指定文字列を直接検索することはできない。そこで、ステップ1905において、前記指定文字列を分解して長さ2の部分文字列からなるリストを生成する。前記のとおり指定文字列が“ガード”だった場合、抽出される部分文字列は“ガー”および“ード”となる。

【0328】この結果、前記ステップ1907において、5803に示す展開済み検索条件データ、すなわち文脈識別子集合が{|C1|}、指定文字列が“ガード”、部分文字列リストが{|“ガー”, “ード”|}であるデータが生成される。

【0329】以上説明したように、本実施例に示した構成によると、複数の構造を持つ文書の検索を一括して行なうことができる。また、検索条件において、種別および要素型名を指定した構造条件が可能となる。

【0330】以上が、本発明を利用した第五の実施例の説明である。

【0331】(第六の実施例)次に、本発明を利用した第六の実施例を図面を利用して説明する。

【0332】第六の実施例において第五の実施例と異なる点は、文書登録の処理では、種別定義テーブルを利用せず、要素型名をそのまま構造の種別としてメタ構造インデックスを作成し、検索処理の際に、種別を用いた構造

条件を含む構造指定検索条件を、要素型名を指定した構造条件に変換した上で、検索処理を実施する点である。これにより、種別による構造条件と要素型名による構造条件のいずれも指定可能である。

【0333】本実施例における文書登録サブシステム101のシステム構成は、図48に示す第五の実施例の文書登録サブシステム101のシステム構成と同じである。

【0334】ただし、第五の実施例で図48に示したメタ構造インデックス作成プログラム4801のうち、図52を用いて説明した、ステップ5201とステップ5202の処理内容を一部変更する。すなわち、ステップ5201とステップ5202において、種別定義テーブル4803を参照して要素型名から種別への変換を行う処理は実行せず、要素型名をそのまま種別とみなして、構造インデックスへの登録処理を行う。ただし、種別定義テーブル4803を作成し、検索サーバへ転送する処理は変わらない。

【0335】以上が、第六の実施例における文書登録サブシステム101の、第五の実施例における文書登録サブシステムからの変更点である。その他の構成、処理内容は全て同じである。

【0336】本実施例における文書検索サーバ102のシステム構成も図55と同じである。

【0337】ただし、第五の実施例で図55に示したメタ構造インデックス作成プログラム5501のうち、図57を用いて説明したステップ5702の処理が一部変更される。

【0338】つまり、ステップ5702において、検索条件で指定された構造条件が、構造の種別により指定されていれば、種別定義テーブル4803を参照して、該種別に対応する全ての要素型名を取得し、得られた要素型名のORにより作成した構造条件に変更する。メタ構造インデックスを辿ることで、作成した構造条件を満たす構造内に含まれる文字列データの文脈識別子を得ることができる。要素型名で指定された場合は、そのまま、メタ構造インデックスを辿って、構造条件を満たす構造内に含まれる文字列データの文脈識別子を得る。

【0339】図59に、本実施例のステップ5702において、種別による構造条件が指定された場合に要素型名に変更して生成する構造条件を示す。本図に示すように、構造条件を構成する種別ごとに、要素型名を取得し、構造条件の各階層を1つあるいは複数の要素型名の論理和(以下OR)で記述する構造条件を生成する。複数の要素型名のORは、構造条件に「[種別または要素型名, 種別または要素型名, …]」のように「|」内に複数の種別または要素型名を列挙するなどの方法により指定する。

【0340】ユーザが検索条件を種別を指定する際には、種別名の前に“Type:”などの識別情報を記載するこ

57

とで、その名称を種別として、要素型名に変更する。何も指定せずに検索条件を記述する場合は、要素型名であると判断し、そのまま、構造条件を利用して、構造インデックスから適合する構造の文脈識別子を取得する。あるいは、同じ名称の要素型名が存在しない種別であれば、曖昧性がないため、"Type:"などの識別指定を省略してもよい。

【0341】図59では、「Type:属性/Type:題目」という構造条件(5901)を、構造条件の変換処理(5902)により、種別定義テーブル4803を参照して、要素型名を用いた構造条件(5903)に変換する。

【0342】検索クライアント103、104の変更点は、第五の実施例で図58を用いて説明した通りである。ただし、本実施例では、要素型名を指定した構造条件は、要素型名が一致する構造内の文字列データの文脈識別子が取得でき、種別を指定した構造条件は、種別が一致する構造内の文字列データの文脈識別子が取得できる。

【0343】以上の処理により、本実施例では、検索条件に種別による構造条件と要素型名による構造条件を組み込むことが可能となる。

【0344】さらに、本実施例の方法の第五の実施例の方法と比較したメリットは、構造インデックスが、要素型名毎にメタノードが作成されるため、種別の変更が任意に可能であるということである。例えば、クライアント毎に種別定義テーブルを作成して検索サーバに転送した後に、種別定義テーブルを指定した検索条件を設定するなどの処理方法も可能であり、柔軟な種別の設定が実現できる。第五の実施例では、種別定義テーブルの変更に

対応するためには、変更時点までに作成した、メタ構造インデックス、文字列インデックスの再作成が必要となる。

【0345】また、本実施例の方法のデメリットとしては、第五の実施例に比べて、生成されるメタ構造インデックスが大きくなることがある。メタ構造インデックスにおいて、種別ごとにメタノードを作成する方が、要素型名ごとに作成する場合に比べて、メタノード数を削減できるためである。

【0346】以上が、本発明を利用した第六の実施例の説明である。

【0347】(第七の実施例)次に第七の実施例として、異なる文書構造を持つ文書群を、メタ構造インデックスを使わずに1つの構造インデックスを利用して構造指定検索するためのシステム構成および処理手順を説明する。

【0348】図60は、本実施例における文書登録サブシステム101のシステム構成である。本システム構成は、そのハードウェア構成においては、図2に示す第一の実施例の文書登録サブシステム101と変わらない。ただし、主メモリ207に格納する文書登録プログラム

58

にルートノード付加プログラム6001を追加する。

【0349】ルートノード付加プログラム6001の処理内容を図61を用いて説明する。ルートノード付加プログラム6001は、文書構造解析プログラム210の出力結果である解析済み文書データ6101を解析済み文書データ格納領域217から読み出し、該解析済み文書データの最上位ノードの上位ノードとして、固定の種別を持つノードを付加した、ルートノード付加解析済み文書データ6102を作成し、解析済み文書データ格納領域217に格納する。これにより、読み出した解析済み文書データ6101をルートノード付加解析済み文書データ6102に置き換える。

【0350】本実施例の文書登録サブシステム101において、上記の処理以外は、第一の実施例の文書登録サブシステムの構成および処理と全て同じである。

【0351】図62は、本実施例における文書検索サーバ102のシステム構成である。本システム構成は、そのハードウェア構成においては、図17に示す第一の実施例の文書検索サーバ102と変わらない。ただし、主メモリ207に格納する文書登録プログラムに検索条件修正プログラム6201を追加する。

【0352】検索条件修正プログラム6201は、構造条件が最上位構造から指定された場合は、文書登録時に登録文書の解析済み文書データの最上位構造に追加したrootを構造条件に追加する処理が加わる。その他の場合は、検索条件を変更することはない。

【0353】図63は、検索条件修正プログラム6201の処理内容を示すPAD図である。

【0354】まず、ステップ6301で、検索条件中の構造条件の有無をチェックする。構造条件が存在するなら、ステップ6302に進み、存在しないなら、検索条件を何も変更せず、検索条件修正プログラム6201を終了する。

【0355】ステップ6302で、構造条件が最上位構造から指定されているか否かをチェックする。最上位構造から指定されている場合は、ステップ6303に進む。最上位構造から指定されていない場合は、検索条件を何も変更せず、検索条件修正プログラム6201を終了する。

【0356】ステップ6303で、構造条件を変更し、最上位構造のrootを指定した検索条件とする。

【0357】ステップ6304で、変更した検索条件を出力する。1702以降の処理内容は、図17を用いて前述した第一の実施例の検索サーバ102の処理内容と同じである。

【0358】図64は、検索条件修正の結果である。本図に示すように構造条件の構造条件に、最上位構造が指定される場合は、rootという構造を追加した構造条件を生成する。

【0359】上記の処理により検索条件を変更する処理

以外は、全て第一の実施例の検索サーバ 102 の構成および処理内容と同じである。

【0360】次に、第七の実施例における検索クライアント 103、104 の処理内容について説明する。

【0361】本実施例において、検索クライアント 103、104 のシステム構成は、図 25 の第一の実施例の検索クライアントのシステム構成と同じである。

【0362】ただし、検索結果表示プログラム 2502 の処理手順を示す図 28 の PAD 図のステップ 2815 において、解析済み文書データを書式化して表示する際に、文書登録サブシステム 101 において、追加されたルートノードは削除してから書式化して表示する。つまり、登録文書を構造解析した結果である解析済み文書データに変換してから表示するように変更する。これにより、ユーザからは、解析済み文書データに追加されたルートノードの存在は見えないことになる。

【0363】以上が第七の実施例における、第一の実施例の処理からの変更点である。その他の構成および処理内容は、第一の実施例の構成および処理内容と同じである。

【0364】以上の処理により、解析済み文書データが登録文書と異なる点を除いて、構造インデクスを利用してメタ構造インデクスを利用した場合と同様に様々な文書構造を持つ文書群に対する一括した構造指定検索が可能となる。

【0365】（第八の実施例）次に、複数の構造において同じ種別を持つ構造をまとめて効率的に検索するための別名構造インデクスの作成方法およびこれを用いた検索処理について説明する。

【0366】図 65 は、別名構造インデクス 6501 の構成および別名構造インデクス 6501 とメタ構造インデクス 6502 との関係および、別名構造インデクスを作成するために用いる別名定義テーブル 6503 の内容を示した図である。

【0367】別名構造インデクスは、必ずしも構造インデクスのように文書全体の構造を辿るためのインデクスを作成するわけではなく、文書構造の部分構造を構造インデクスから切り出し、切り出した部分構造を重ね合わせて作成する。

【0368】図 65 に示すように、異なる文書構造の書誌に関する情報を切り出し、メタ構造インデクスを構成するメタノードの文脈識別子を管理することで、検索条件として個々の構造を指定しなくても、1 つの別名を指定した構造条件を設定することで、その別名に対応する全てのメタ構造インデクス中のメタノードの文脈識別子を得ることができる。

【0369】別名定義テーブル 6503 には、本図で示したように、別名 6504 と構造定義数 6505 と構造定義 6506 が格納される。

【0370】別名 6504 は、別名構造インデクスを参

照する際の名称が格納される。構造定義数 6505 には、別名登録された構造定義数を記述する、構造定義 6506 には、別名 6504 で代表される検索条件中の構造条件を構造定義数分列挙する。

【0371】別名構造インデクスは、いくつかの構造定義によって指定される構造インデクス中のメタノードの文脈識別子をあらかじめ取得しておき、構造条件から検索条件を満たす構造に含まれる文字列データの文脈識別子を高速に取得するためのインデクスである。

【0372】別名構造インデクスの各ノードには、構造インデクスのメタノードと同じように、論理構造をあらわすためのリンク情報とメタノードの文脈識別子を持つ。ただし、メタノードの文脈識別子には、別名として定義された構造に含まれる文字列データのメタノードの文脈識別子をすべて格納する。

【0373】図 66 は、本実施例における文書登録サブシステム 101 のシステム構成を示す図である。

【0374】本実施例における文書登録サブシステム 101 のシステム構成は、図 48 に示す第五の実施例の文書登録サブシステム 101 のシステム構成とそのハードウェア構成に関しては変わるところはない。

【0375】ただし、主メモリ 207 に格納する文書登録プログラムに別名構造インデクス作成プログラム 6601 を追加し、磁気ディスク 208 には、別名構造インデクス格納領域 6602 および別名定義テーブル 6603 を追加する点が異なる。

【0376】別名構造インデクス作成プログラム 6601 は、別名定義テーブル格納領域 6603 から別名定義テーブルを読み出す。さらに、メタ構造インデクス作成プログラム 4801 で作成するメタ構造インデクスをメタ構造インデクス格納領域 4802 から読み出す。読み込んだ情報を元に、別名構造インデクスを作成し、別名構造インデクス格納領域 6602 に格納する。

【0377】図 67 は、本実施例における文書登録サブシステム 101 の概略処理手順を示す PAD 図である。本実施例の処理手順は、図 51 を用いて前述した、第五の実施例の文書登録サブシステム 101 の概略処理手順とはほぼ同じである。ただし、ステップ 5101 のあとで、ステップ 6701 を実行する点と、ステップ 5102 の代わりにステップ 6702 を実行する点が異なる。

【0378】ステップ 6701 では、別名構造インデクス作成プログラム 6601 を実行し、文書の登録により更新されたメタ構造インデクスの情報を参照し、別名構造インデクスの内容を更新する。

【0379】ステップ 6702 では、全ての解析済み文書データ、メタ構造インデクス、別名構造インデクス、文字列インデクスを文書検索サーバ 102 に転送する。

【0380】図 68 は、図 67 のステップ 6701 の詳細手順を示す PAD 図である。本図を用いて、別名構造インデクスの作成手順を説明する。

61

【0381】まず、ステップ6801で、別名として作成する構造を定義した別名定義テーブル6603を読み出す。別名定義テーブル6603は、ユーザがキーボード202からテキストエディタなどを利用して作成する。あるいは、構造インデクスから、異なる階層に存在する同じ種別の構造を抽出して、この情報を元に別名定義テーブルを作成するプログラムによって作成する。

【0382】次に、ステップ6802で、上記ステップ6801で読み出した別名定義テーブル6603を用いて、構造インデクス中から、該構造情報に適合するメタノードを抽出する。これは、図57の5702ステップで前述した、第五の実施例における文書検索処理における検索条件に適合するメタ構造インデクスの取得処理と同じ処理で実現できる。

【0383】ステップ6803では、得られたメタノードの文脈識別子を管理するテーブルを作成し、これを別名構造インデクスに登録する。

【0384】ステップ6804では、階層構造がある別名について、階層構造を表現するためにノード同士を接続する。別名構造インデクスに登録する別名は、“書誌/題目”のように階層的な別名を指定することが可能である。この場合、まず、構造インデクスから、“書誌”の種別情報を持つメタノードを抽出した上で、その子ノードから“題目”の種別情報を持つメタノードを抽出し、このメタノードの文脈識別子を管理する文脈識別子管理テーブルを作成して、別名構造インデクスに登録する。さらに、この過程で得られる“書誌”の種別情報を持つメタノードについても文脈識別子管理テーブルを作成し、別名構造インデクスの“書誌”に格納することで、階層的な構造を持った、別名構造インデクスを作成する。

【0385】図69に、本実施例における全文検索サーバ102のシステム構成図を示す。

【0386】本構成図は、図55を用いて前述した第五の実施例における全文検索サーバ102のシステム構成図とそのハードウェア構成においては同じである。ただし、主メモリ207にメタ構造インデクス対応検索条件解析プログラム5501の代わりに別名構造インデクス対応検索条件解析プログラム6901を格納する点と、磁気ディスク208に別名構造インデクス格納領域を追加する点異なる。

【0387】図70は、本実施例における検索処理の概要を示すPAD図である。

【0388】本図に示した処理は、図56に示した第五の実施例の検索処理とほぼ同じであるが、ステップ5601の代わりにステップ7001を実行する点とステップ5602の代わりにステップ7002を実行する点異なる。

【0389】ステップ7001では、メタ構造インデクス、別名構造インデクス、文字列インデクスを文書登録サブシステム101から受信して、それぞれメタ構造イ

62

ンデクス格納領域4802、別名構造インデクス格納領域6602、文字列インデクス格納領域220に格納する。

【0390】ステップ7002では、別名構造インデクス対応検索条件解析プログラム6901を実行する。

【0391】図71は、ステップ7002の処理の詳細すなわち、別名構造インデクス対応検索条件解析プログラム6901の処理手順を示すPAD図である。

【0392】本図の処理は、図57に示す第五の実施例のメタ構造インデクス対応検索条件解析プログラムの処理手順とほぼ同じである。

【0393】ただし、構造条件の有無を判定するステップ1901の代わりに、構造条件の有無および別名指定可否かを判定するステップ7101を実行する点と、ステップ7101で別名と判定された場合は、ステップ7102、ステップ7103を実行する点異なる。構造条件が種別もしくは要素型名の場合は、第五の実施例と同じく、ステップ5701とステップ5702を実行する。

【0394】ステップ7101では、構造指定検索の構造条件として、別名を利用しているか否かを判定する。構造条件で別名を利用する場合は、構造条件の先頭に、例えば“Alias:”という文字列を指定することで区別する。したがって、別名である“題目”を検索対象の構造として指定する場合は、“Alias:題目”と構造条件に記述しているか否かをチェックすることで判定することができる。

【0395】ステップ7102では、別名構造インデクスを読み込む。次にステップ7103で、別名構造インデクスを参照し、指定された構造条件を満たす文字列データの文脈識別子の集合を求める。別名インデクスに格納された、別名に対応するメタ構造インデクスのメタノードの下位にある文字列データのメタノードの文脈識別子を取得することで実現できる。

【0396】その他の処理は、図57で示した、第五の実施例のメタ構造インデクス対応検索条件解析プログラムの処理手順と同じである。

【0397】本実施例における、検索サーバ102の構成および処理内容は、その他の点においては、全て第五の実施例における、全文検索サーバ102の処理と同じである。

【0398】以上が、本発明を利用した第八の実施例の説明である。

【0399】（第九の実施例）次に第九の実施例として、第五の実施例における種別定義テーブル4803の記載内容を変更し、各文書構造毎に要素型名の種別を指定できるようにする方法について説明する。

【0400】図72を用いて、本実施例における種別定義テーブル4803の格納情報について説明する。本図のように、DTD名称と要素型名を合わせてDTDおよび要素

10

20

30

40

50

63

型名の領域 7201 に格納することで、解析済み文書データの要素型名だけでなく、DTD 名称との組み合わせで種別を決めることができる。これにより、例えば、「報告書」における「本文」の種別は「報告内容」であり、その他の文書の「本文」は、種別も「本文」のままとし、登録文書の文書構造に合わせた種別を定義することが可能となる。

【0401】本実施例における文書登録サブシステム 101 のシステム構成は、図 48 に示す第五の実施例の文書登録サブシステムのシステム構成と同じである。さらに、本実施例における、文書登録サブシステム 101 の処理手順は、図 52 の PAD 図に示す第五の実施例における文書登録サブシステムの処理手順と同じである。ただし、ステップ 5201 における構造インデクスの最上位構造の取得の際に種別定義テーブル 4803 を参照して単に要素型名を種別に変換するのではなく、登録文書の DTD と要素名の組み合わせに対応する種別を取得する点異なる。さらに、ステップ 5202 において、構造インデクスの重ね合わせを行なう際に、種別定義テーブル 4803 を参照して、登録文書の DTD と要素型名の組み合わせにより種別を得てから重ね合わせを行なう点異なる。

【0402】その他の構成及び処理内容において、本実施例が第五の実施例と異なる点はない。

【0403】以上が、本発明を利用した第九の実施例である。

【0404】（第十の実施例）次に第十の実施例として、第五の実施例における種別定義テーブル 4803 を構造インデクス毎に管理し、メタ構造インデクス中の各構造インデクス毎に種別定義テーブルを参照して種別を得る方法について説明する。

【0405】本実施例における文書登録サブシステム 101 のシステム構成は、図 48 に示す第五の実施例の文書登録サブシステムのシステム構成と同じである。ただし、本実施例では、種別定義テーブル 4803 を各構造インデクス毎に作成し、メタ構造インデクスには、各構造インデクスの最上位構造の種別に関する種別定義テーブルを持つ構成とする点異なる。このような構成とすることで、1つの要素型名を構造インデクス毎に異なる種別に割り当てることができる。

【0406】これらの種別定義テーブルは、第五の実施例において、図 50 に示す内容としても良いし、第九の実施例において、図 72 で示す内容としても良い。以下の説明では、種別定義テーブルは図 50 に示す第五の実施例の種別定義テーブルを用いる場合について説明するが、図 72 で示した第九の実施例の種別定義テーブルを用いる場合でも同様の手順で実現可能である。

【0407】図 73 は、本実施例におけるメタ構造インデクスと、種別定義テーブルの関連を示す図である。メタ構造インデクス 7301 のルートメタノード 7302

64

に対応する、最上位構造種別定義テーブル 7303 を作成する。さらに、各構造インデクス毎に種別定義テーブルを作成する。本図では、論文の構造インデクス 7304 に対応する種別定義テーブル 1 (7305) を作成し、さらに報告書の構造インデクス 7306 に対応する種別定義テーブル 2 (7307) を作成する。このような構成にすることで、構造インデクス毎に種別を定義することができる。

【0408】本実施例における、文書登録サブシステム 101 の処理手順は、図 52 に示す第五の実施例における文書登録サブシステムの処理手順を示した PAD 図と同じである。ただし、ステップ 5201 における構造インデクスの最上位構造の取得の際には、最上位構造種別定義テーブル 7303 を参照して要素型名を種別に変換し、対応する構造インデクスを得る。さらにステップ 5202 において、ステップ 5201 で得られる構造インデクスに対応する種別定義テーブルを参照して、要素型名を種別に変換し構造インデクスの重ね合わせを行なう。例えば登録文書の種別が論文であれば、構造インデクス 7304 に対応する種別定義テーブル 1 (7305) を参照して、要素型名を種別に変換する。

【0409】その他の構成及び処理内容において、本実施例が第五の実施例と異なる点はない。

【0410】第九の実施例のように種別定義テーブルが図 72 に示す構成であっても、第九の実施例で示した手順により要素型名だけでなく DTD 名称との組み合わせで参照することで、同様に実現可能である。

【0411】以上が、本発明を利用した第十の実施例である。

【0412】

【発明の効果】以上説明したように、本発明による構造化文書の検索方法によれば、文書中に現れる論理要素の文書中における出現位置に関する条件を構造条件指定中に含めることができるので、複雑な論理構造を備えた多数の文書からなる文書データベース上においても精度の高い構造指定検索ができる。

【図面の簡単な説明】

【図 1】本発明による文書検索システムの第一の実施例の全体構成を示す図である。

【図 2】本発明の第一の実施例における文書登録サブシステムの構成を示す図である。

【図 3】本発明の第一の実施例における文書登録処理の概略手順を示す PAD 図である。

【図 4】文書論理構造を定義する DTD の一例を示す図である。

【図 5】SGML による構造化文書の記述例を示す図である。

【図 6】SGML が表現する文書の論理構造を図形的に示した模式図である。

【図 7】本発明の第一の実施例における文書構造解析プ

ログラムの処理手順を示すPAD図である。

【図 8】文書構造テーブルのデータ構造を示す図である。

【図 9】本発明の第一の実施例における構造インデックス作成プログラムの処理手順を示すPAD図である。

【図 10】本発明の第一の実施例における解析済み文書データの辿り順を示す図である。

【図 11】本発明の第一の実施例におけるノードとメタノードとの対応関係を示す図である。

【図 12】本発明の第一の実施例における構造インデックスの生成過程を示す図である。

【図 13】本発明の第一の実施例における構造化全文データ生成プログラムの処理手順を示す図である。

【図 14】本発明の第一の実施例における構造化全文データのファイル形式を示す図である。

【図 15】本発明の第一の実施例における文字列インデックス作成プログラムの処理手順を示すPAD図である。

【図 16】本発明の第一の実施例における文字列インデックスのデータ構造を示す図である。

【図 17】本発明の第一の実施例における文書検索サーバの構成を示す図である。

【図 18】本発明の第一の実施例における文書検索処理の概略手順を示すPAD図である。

【図 19】本発明の第一の実施例における検索条件解析プログラムの処理手順を示すPAD図である。

【図 20】本発明の第一の実施例における展開済み検索条件データの生成例を示す図である。

【図 21】本発明の第一の実施例における文字列インデックス検索プログラムの処理手順を示すPAD図である。

【図 22】本発明の第一の実施例における接続判定処理の実行例を示す図である。

【図 23】本発明の第一の実施例における検索結果データのデータ構造を示す図である。

【図 24】本発明の第一の実施例における検索結果データ転送処理の詳細手順を示すPAD図である。

【図 25】本発明の第一の実施例における文書検索クライアントの構成を示す図である。

【図 26】本発明の第一の実施例における検索クライアントの動作手順を示すPAD図である。

【図 27】本発明の第一の実施例における検索条件入力プログラムの処理手順を示すPAD図である。

【図 28】本発明の第一の実施例における検索結果表示プログラムの処理手順を示すPAD図である。

【図 29】本発明の第二の実施例における文書登録システムの構成を示す図である。

【図 30】本発明の第二の実施例における文書登録処理の概略手順を示すPAD図である。

【図 31】本発明の第二の実施例における逆順構造イン

デックス作成プログラムの処理手順を示すPAD図である。

【図 32】本発明の第二の実施例における解析済み文書データの辿り順を示す図である。

【図 33】本発明の第二の実施例におけるノードとメタノードとの対応関係を示す図である。

【図 34】本発明の第二の実施例における逆順構造インデックスの生成過程を示す図である。

【図 35】本発明の第二の実施例における構造化全文データ生成プログラムの処理手順を示すPAD図である。

【図 36】本発明の第二の実施例における構造化全文データのファイル形式を示す図である。

【図 37】本発明の第二の実施例における文字列インデックスのデータ形式を示すPAD図である。

【図 38】本発明の第二の実施例における文書検索サーバの構成を示す図である。

【図 39】本発明の第二の実施例における文書検索処理の概略手順を示すPAD図である。

【図 40】本発明の第二の実施例における検索条件解析プログラムの処理手順を示すPAD図である。

【図 41】本発明の第二の実施例における展開済み検索条件データの生成例を示す図である。

【図 42】本発明の第二の実施例における文字列インデックス検索プログラムの処理手順を示すPAD図である。

【図 43】本発明の第三の実施例におけるノードとメタノードとの対応関係を示す図である。

【図 44】本発明の第三の実施例における文書登録システムの構成を示す図である。

【図 45】本発明の第四の実施例における文書登録処理の概略手順を示すPAD図である。

【図 46】本発明の第四の実施例における文書構造正規化プログラムの処理手順を示すPAD図である。

【図 47】本発明の第四の実施例における正規化処理の具体例を示す図である。

【図 48】本発明の第五の実施例における文書登録システムの構成を示す図である。

【図 49】本発明の第五の実施例におけるメタ構造インデックス作成の例を示す図である。

【図 50】本発明の第五の実施例における種別定義テーブルの内容を示す図である。

【図 51】本発明の第五の実施例における文書登録処理の概略手順を示すPAD図である。

【図 52】本発明の第五の実施例におけるメタ構造インデックス作成処理の概略手順を示すPAD図である。

【図 53】本発明の第五の実施例におけるメタ構造インデックスの更新処理の例 1 を示す図である。

【図 54】本発明の第五の実施例におけるメタ構造インデックスの更新処理の例 2 を示す図である。

【図 55】本発明の第五の実施例における文書検索サー

67

バの構成を示す図である。

【図 56】本発明の第五の実施例における文書検索処理の概略手順を示す P A D 図である。

【図 57】本発明の第五の実施例におけるメタ構造インデクス対応検索条件解析プログラムの処理手順を示す P A D 図である。

【図 58】本発明の第五の実施例における展開済み検索条件データの生成例を示す図である。

【図 59】本発明の第六の実施例における構造条件変換の例を示す図である。

【図 60】本発明の第七の実施例における文書登録サブシステムの構成を示す図である。

【図 61】本発明の第七の実施例におけるルートノード付加プログラムの処理結果の例を示す図である。

【図 62】本発明の第七の実施例における文書検索サーバの構成を示す図である。

【図 63】本発明の第七の実施例におけるルートノード付加プログラムの処理手順を示す図である。

【図 64】本発明の第七の実施例における構造条件変換処理の内容を示す図である。

【図 65】本発明の第八の実施例における別名構造インデクスを示す図である。

【図 66】本発明の第八の実施例における文書登録サブシステムのシステム構成図である。

【図 67】本発明の第八の実施例における登録処理の概略手順の P A D 図である。

【図 68】本発明の第八の実施例における別名構造インデクス作成処理手順を示す P A D 図である。

【図 69】本発明の第八の実施例における文書検索サーバのシステム構成図である。

【図 70】本発明の第八の実施例における文書検索の概略処理を示す P A D 図である。

【図 71】本発明の第八の実施例における別名構造インデクス対応検索条件解析プログラムの処理手順を示す P A D 図である。

【図 72】本発明の第九の実施例における種別定義テーブルの内容を示す図である。

【図 73】本発明の第十の実施例におけるメタ構造インデクスと構造インデクス毎の種別定義管理テーブルの対応関係を示す図である。

【符号の説明】

101…文書登録サブシステム、
102…文書検索サーバ、
103…文書検索クライアント、
210…文書構造解析プログラム、
211…構造インデクス作成プログラム、
212…構造化全文データ生成プログラム、
213…文字列インデクス作成プログラム、
214…文書登録制御プログラム、
217…解析済み文書データ格納領域、

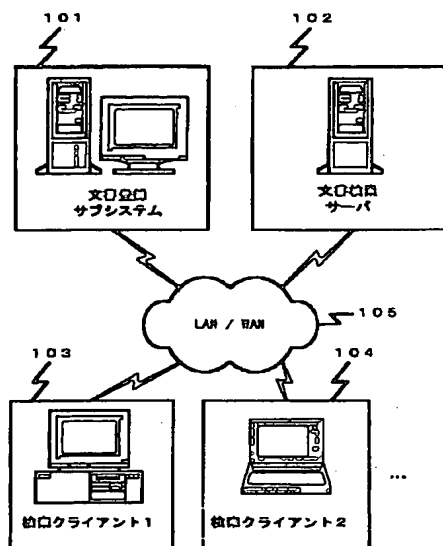
68

218…構造インデクス格納領域、
219…構造化全文データ格納領域、
220…文字列インデクス格納領域、
1701…検索条件解析プログラム、
1702…文字列インデクス検索プログラム、
1703…文書検索制御プログラム、
1704…検索結果データ格納領域、
2001…検索条件の一例、
2002…構造インデクスの一例、
2003…展開済み検索条件データの一例、
2201…文字列インデクスの一例、
2501…検索条件入力プログラム、
2502…検索結果表示プログラム、
2503…クライアント制御プログラム、
2901…逆順構造インデクス作成プログラム、
2902…逆順構造インデクス格納領域、
4102…逆順構造インデクスの一例、
4401…文書構造正規化プログラム、
4801…メタ構造インデクス作成プログラム、
4802…メタ構造インデクス格納領域、
4803…種別定義テーブル、
4901…構造インデクスの一例、
4902…構造インデクスの一例、
4903…ルートメタノード、
4904…メタ構造インデクスの一例、
5501…メタ構造インデクス対応検索条件解析プログラム、
5801…検索条件の一例、
5802…メタ構造インデクスの一例、
5803…展開済み検索条件データの一例、
6001…ルートノード付加プログラム、
6101…解析済み文書データの一例、
6102…ルートノード付加解析済み文書データの一例、
6201…検索条件修正プログラム、
6501…別名構造インデクスの例、
6502…メタ構造インデクスの例、
6503…別名定義テーブルの例、
6601…別名構造インデクス作成プログラム、
6602…別名構造インデクス格納領域、
6901…別名構造インデクス対応検索条件解析プログラム、
7301…メタ構造インデクスの例、
7302…ルートメタノード、
7303…最上位構造種別定義テーブル、
7304…構造インデクス1、
7305…種別定義テーブル1、
7306…構造インデクス2、
7307…種別定義テーブル2。

50

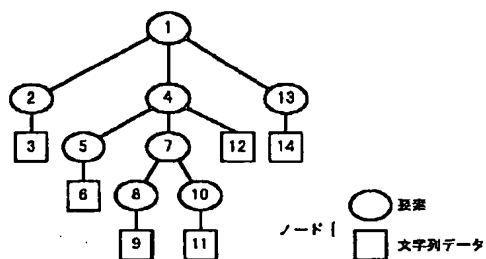
【図 1】

図 1



【図 10】

図 10



【図 50】

図 50

和別定義テーブル(4803)

和別	和別型名	和別型名
属性	3	属性 属性 属性 属性 ...
日付	2	発行日 報告日
題目	4	タイトル 題目 ...
:		

5001

5002

5003

【図 4】

図 4

<!ELEMENT 論文 (タイトル, 執筆者, 日付, 本文, 文献リスト)>

<!ELEMENT タイトル (#PCDATA)>
 <!ELEMENT 執筆者 (名前+)>
 <!ELEMENT 名前 (#PCDATA)>
 <!ELEMENT 日付 (#PCDATA)>

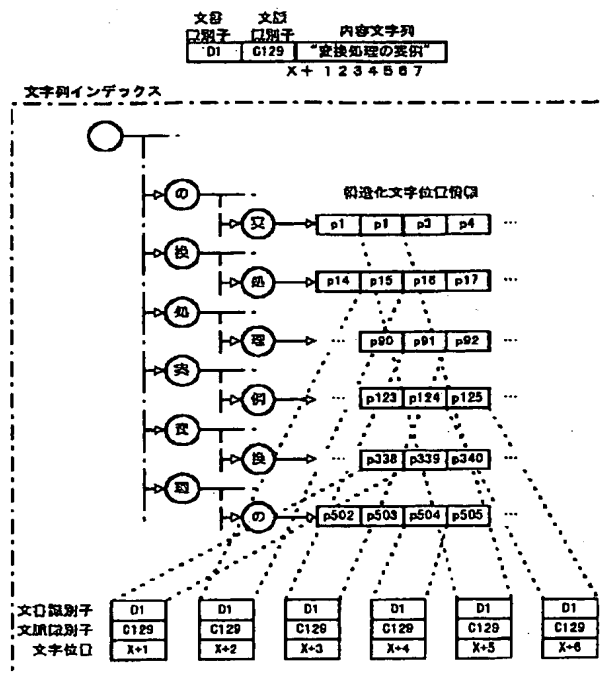
<!ELEMENT 本文 (章*)>
 <!ELEMENT 章 (章題, (段落|備考)*, 節*)>
 <!ELEMENT 章題 (#PCDATA)>
 <!ELEMENT 節 (節題, (段落|備考)*, 項*)>
 <!ELEMENT 節題 (#PCDATA)>
 <!ELEMENT 項 (項題, (段落|備考)*)>
 <!ELEMENT 項題 (#PCDATA)>
 <!ELEMENT 段落 (#PCDATA|強調)*>
 <!ELEMENT 備考 (#PCDATA|強調)*>

<!ATTLIST type (参考|注釈) 参考>
 <!ELEMENT 強調 (#PCDATA)>

<!ELEMENT 文献リスト (文献+)>
 <!ELEMENT 文献 (タイトル, 執筆者, 出典)>
 <!ELEMENT 出典 (#PCDATA)>

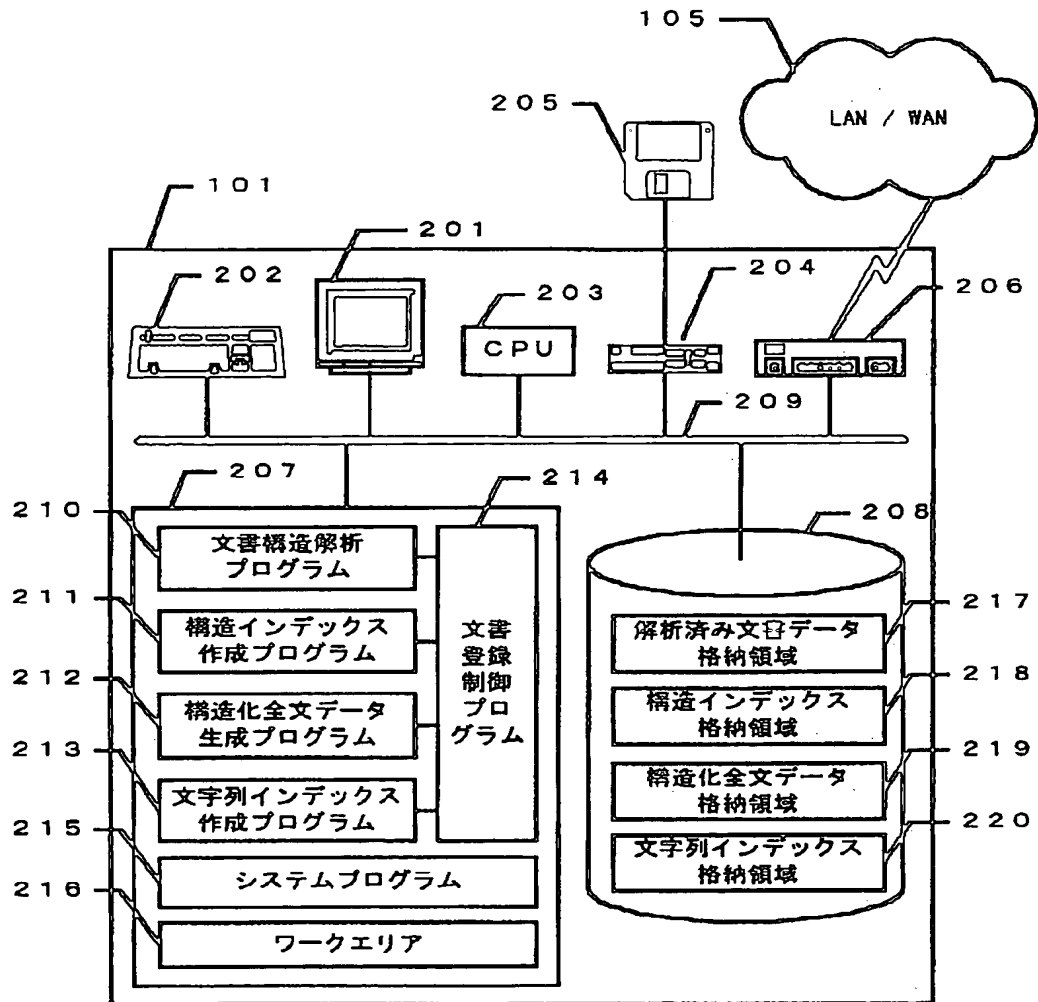
【図 16】

図 16



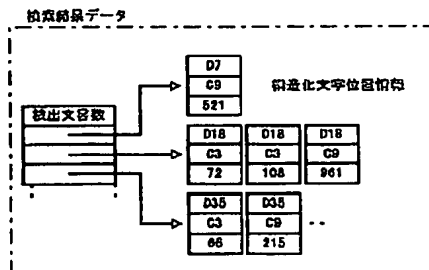
【図 2】

図 2



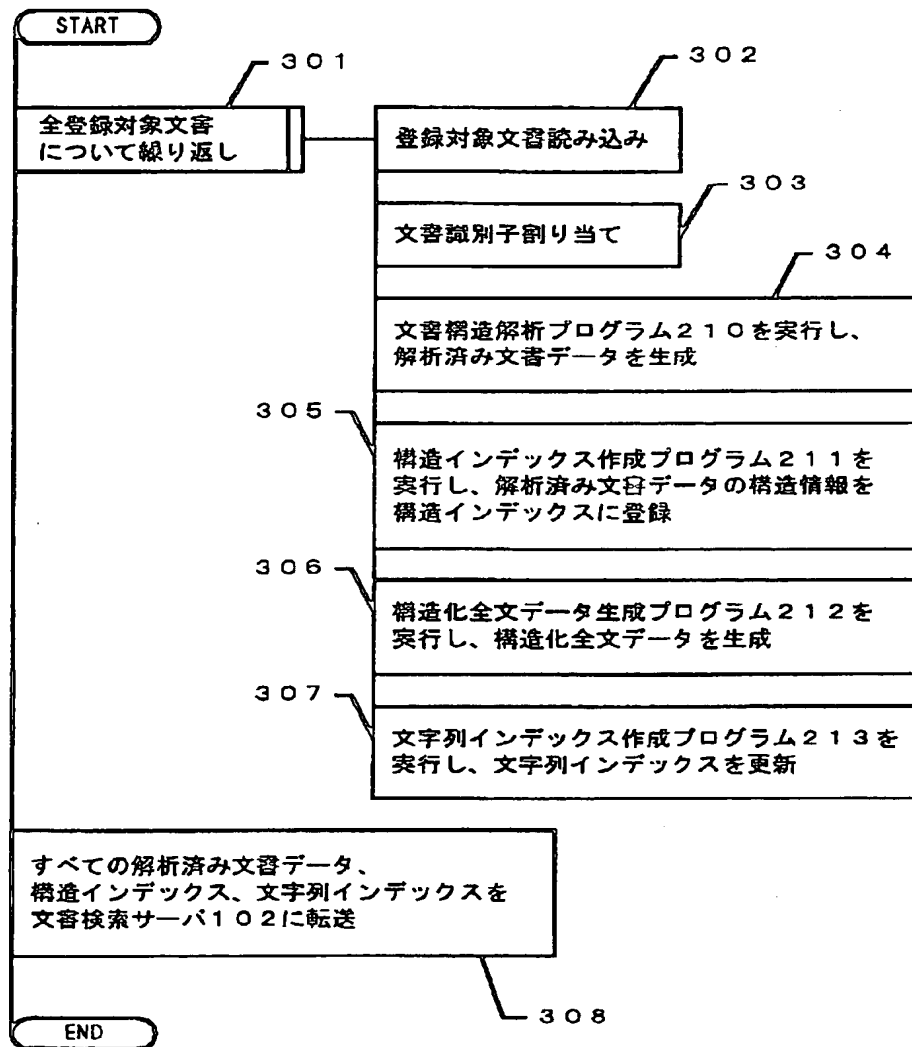
【図 2 3】

図 2 3



【図3】

図 3



【図5】

図5

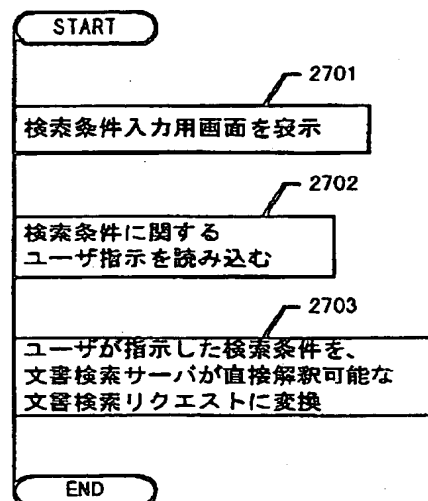
```

<!DOCTYPE 論文 SYSTEM "ronbun.dtd">
<論文>
<タイトル>SGML文書変換言語の開発とその適用事例</タイトル>
<執筆者>
<名前>高橋亨</名前>
<名前>東野純一</名前>
<名前>星幸雄</名前>
</執筆者>
<日付>1996年10月23日</日付>
<本文>
<章>
<章題>はじめに</章題>
<段落>文書記述にSGMLを用いることによって…</段落>
<段落>作成したSGML文書をさまざまな…</段落>…
</章>…
<章>
<章題>適用事例</章題>
<節>
<節題>背景</節題>
<段落>現在、ISOでは…</段落>…
</節>…
<節>
<節題>変換処理の事例</節題>
<項>
<項題>数式の変換</項題>
<段落>JIS規格DTDでは、基本的には数式を…</段落>…
<備考 type=注釈>ただし、行列式の場合には…</備考>…
</項>…
</節>
</章>
</本文>
<文献リスト>
<文献>
<タイトル>SGMLインスタンスの変換方式の検討</タイトル>
<執筆者>
<名前>今郷詔</名前>…
</執筆者>
<出典>情報処理学会第49回全国大会</出典>
</文献>…
</文献リスト>
</論文>

```

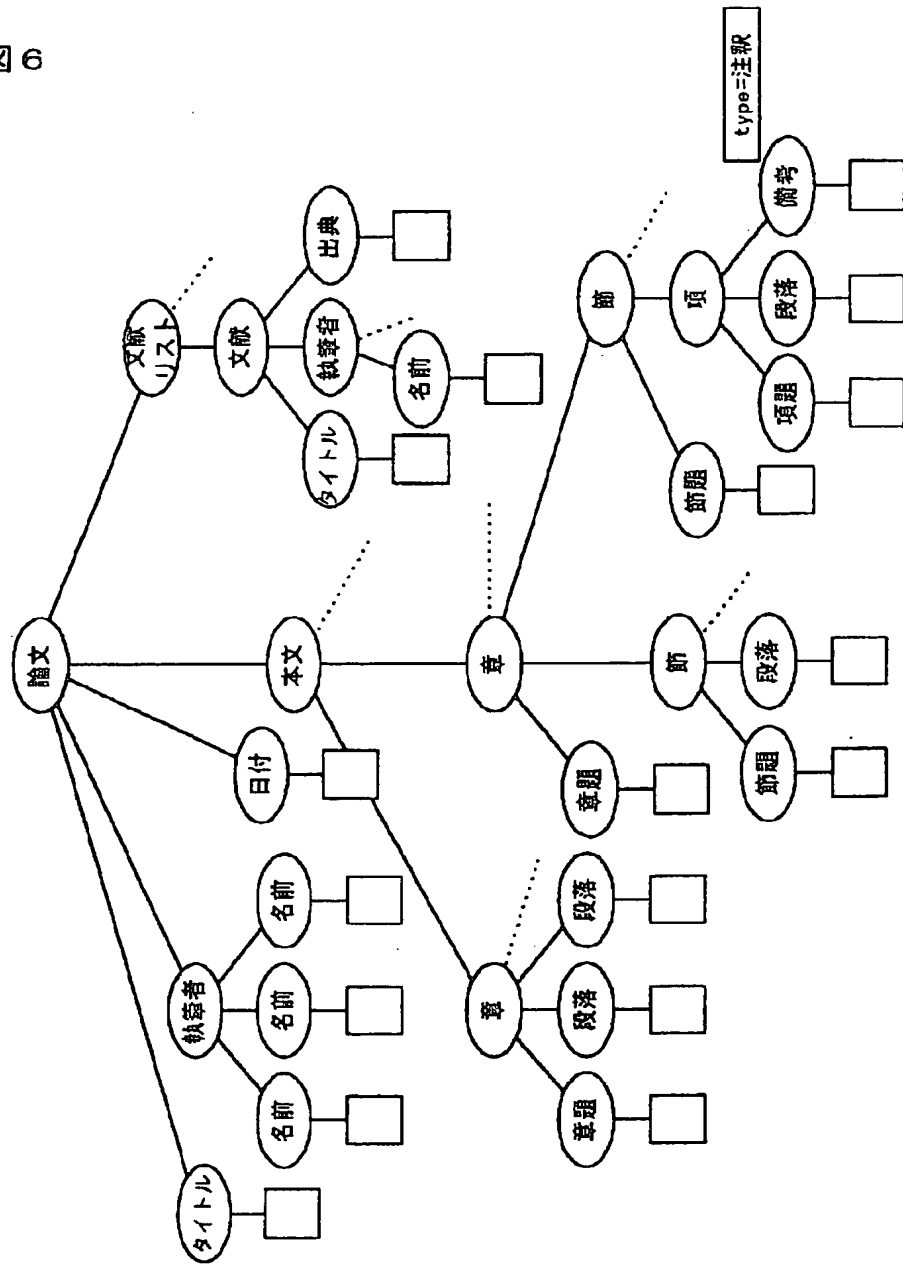
【図27】

図27



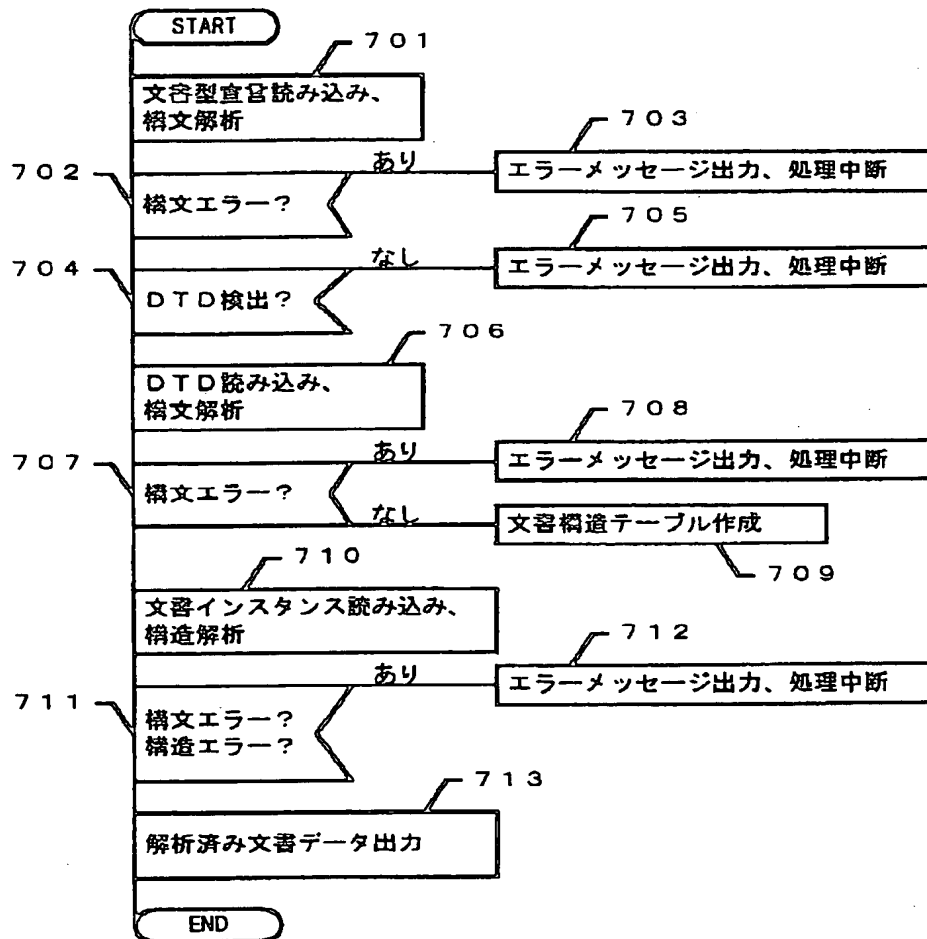
【図 6】

図 6



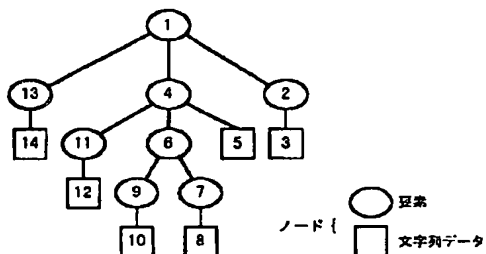
【図 7】

図 7



【図 3 2】

図 3 2



【図 7 2】

図 7 2

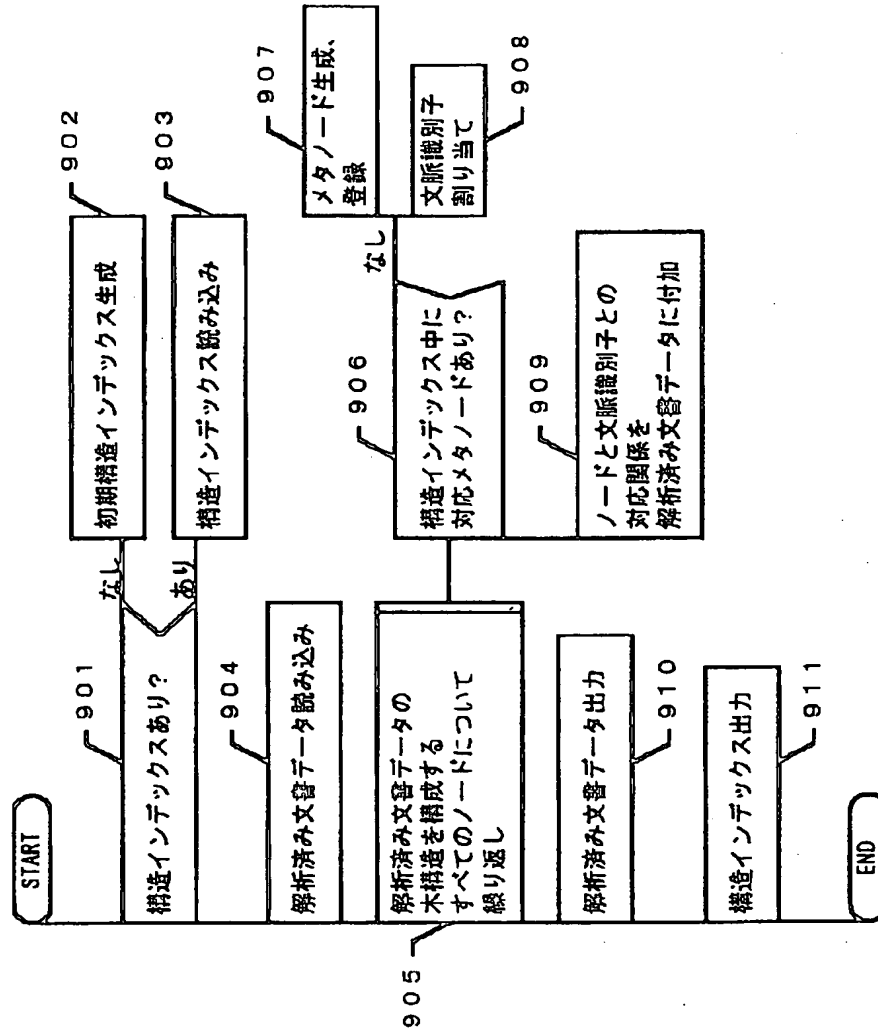
個別定義テーブル(4803)

記号	記号名値	DTD名称および記号名
属性	3	属性名, dtd:属性 属性名, dtd:属性 ...
内容	1	内容名, dtd:本文
本文	3	本文名, dtd:本文 ...
:		

5001 5002 7201

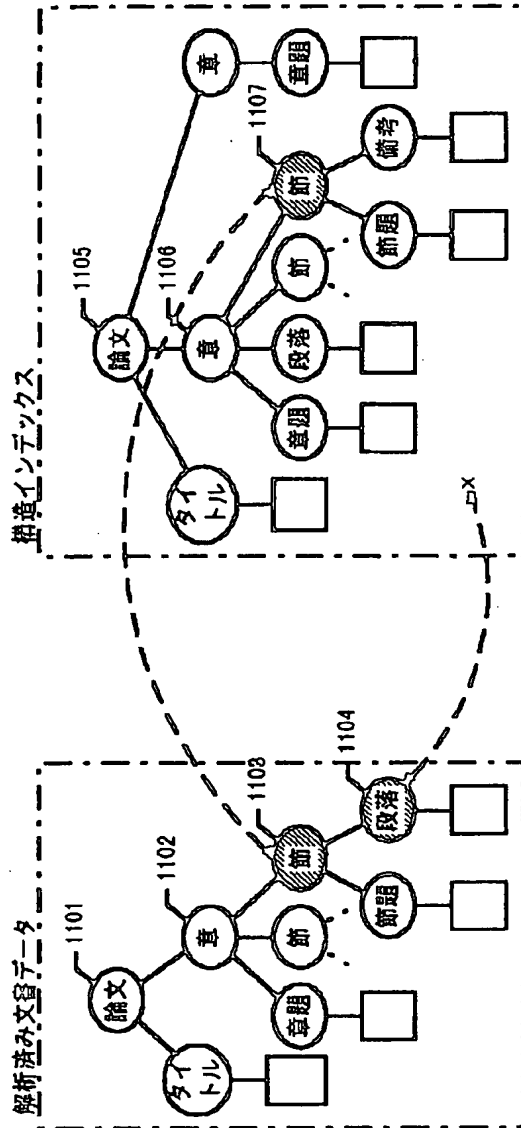
【図 9】

図 9



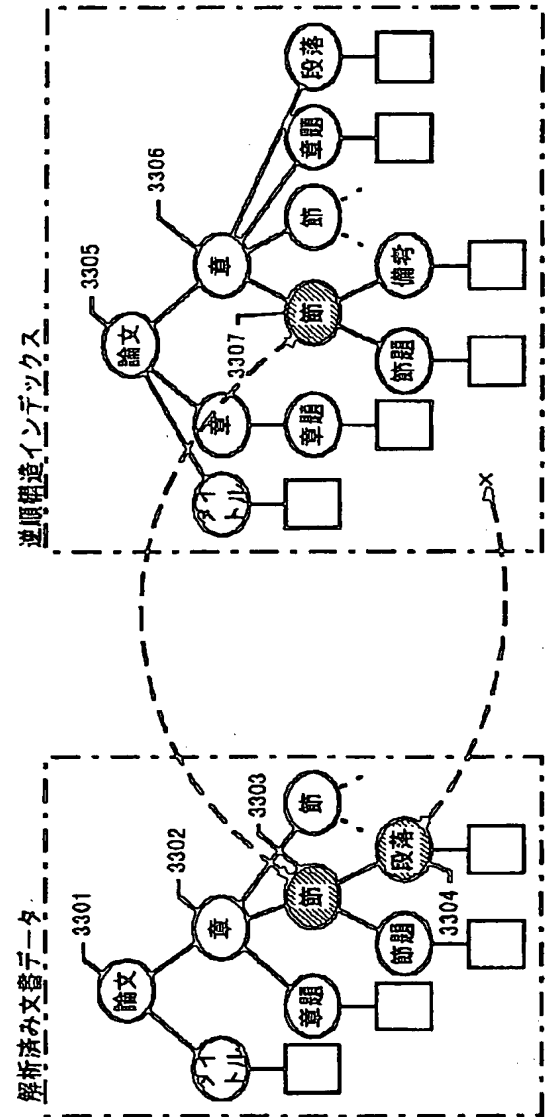
【図 1 1】

図 1 1



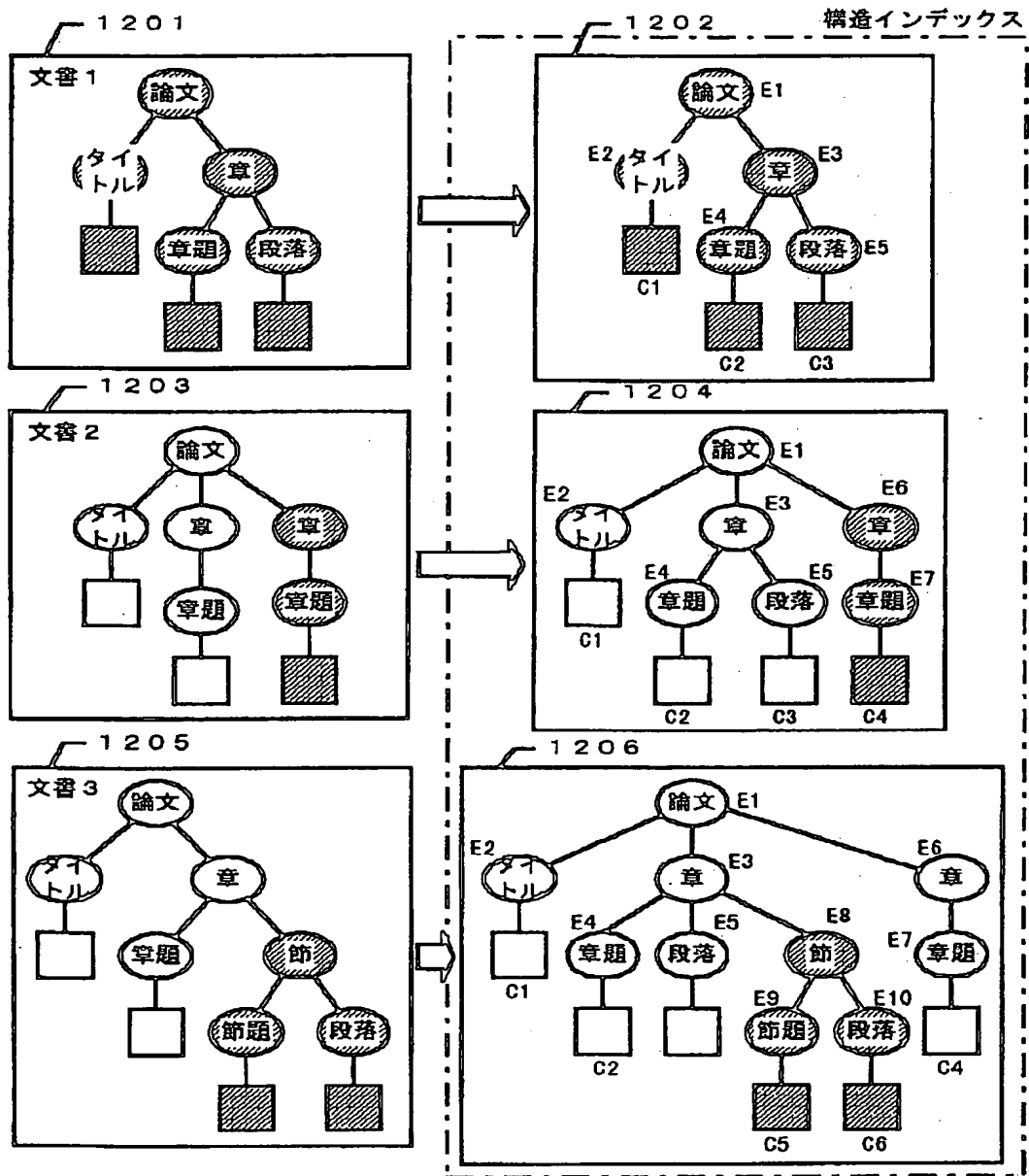
【図 3 3】

図 3 3



【図 12】

図 12



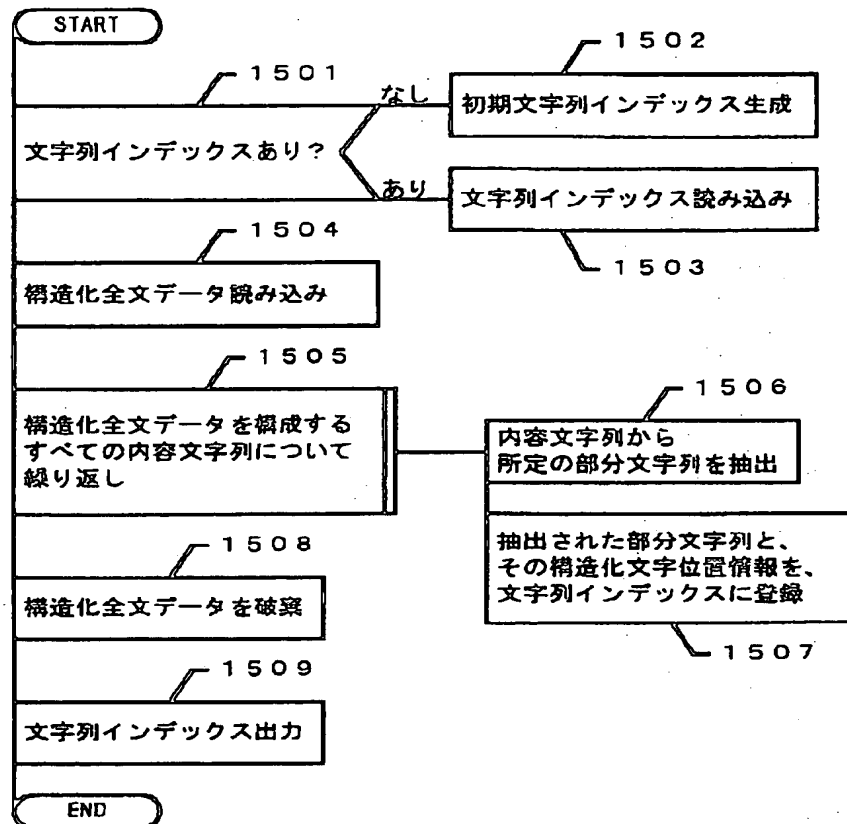
【図 1 4】

図 1 4

文番 識別子	文脈 識別子	内容文字列
D1	C1	“SGML文書変換言語の開発とその適用事例”
	C2	“高橋亨”
	C3	“東野純一”
	C4	“星幸雄”
	C5	“1996年10月23日”
	C6	“はじめに”
	C7	“文番記述にSGMLを用いることによって…”
	C8	“作成したSGML文書をさまざまな…”
⋮		
D1	C15	“適用事例”
	C16	“背景”
	C17	“現在、ISOでは…”
⋮		
D1	C129	“変換処理の実例”
	C130	“数式の変換”
	C131	“JIS規格DTDでは、基本的には数式を…”
	C132	“ただし、行列式の場合には…”
⋮		

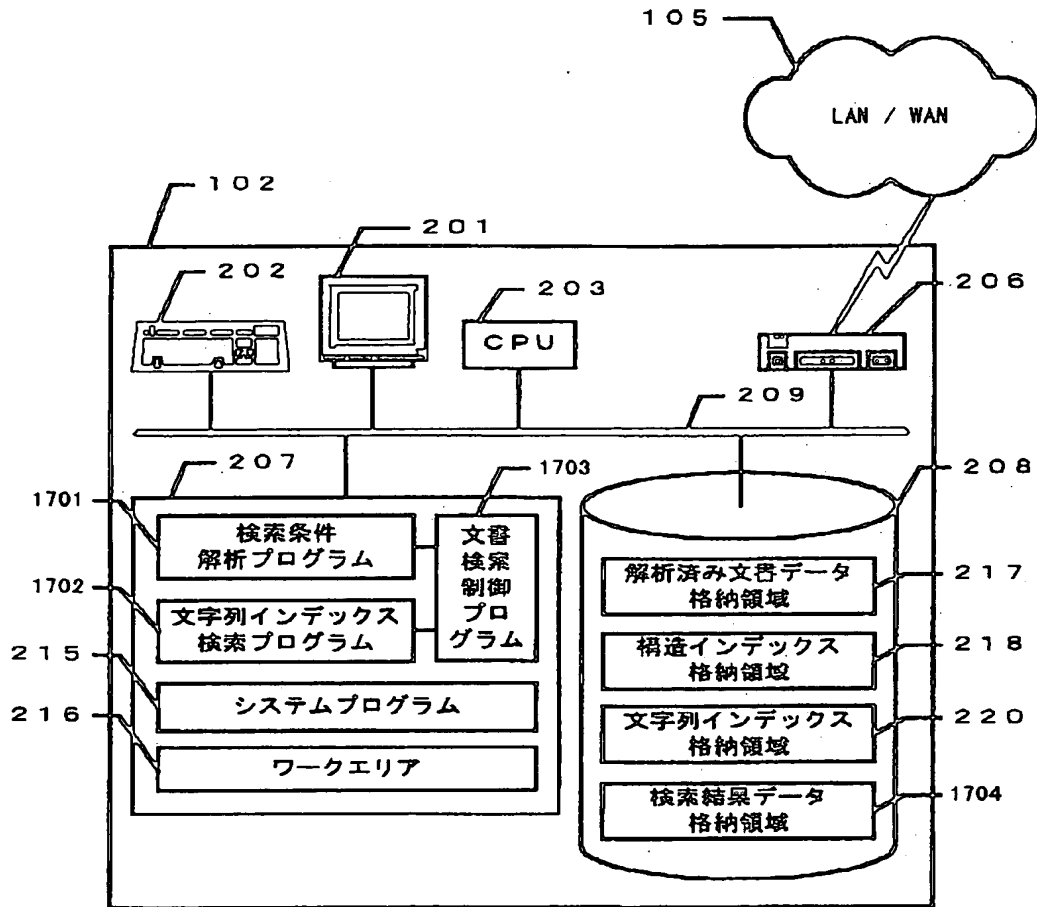
【図 15】

図 15



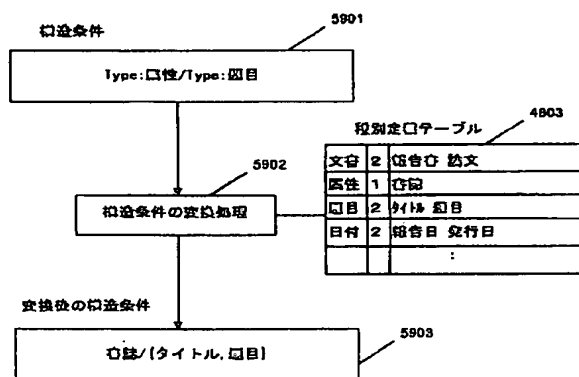
【図17】

図17



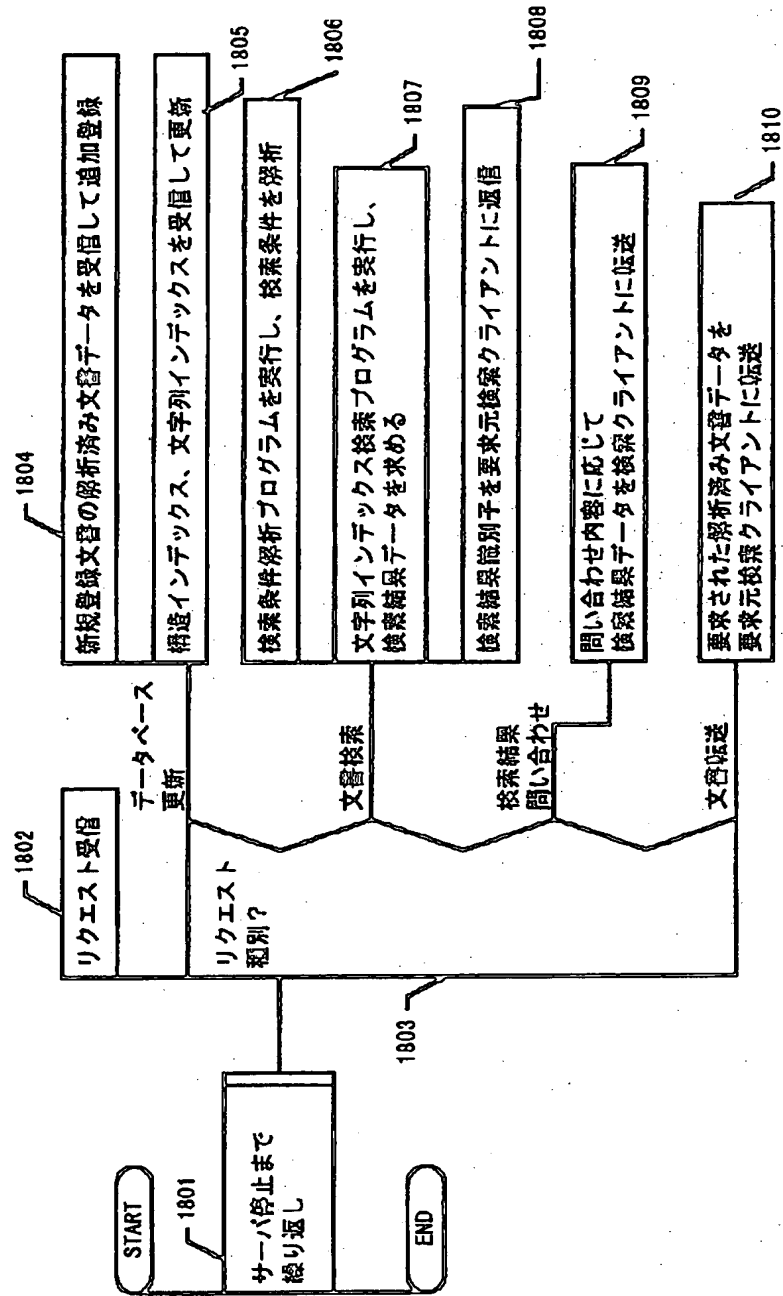
【図59】

図59



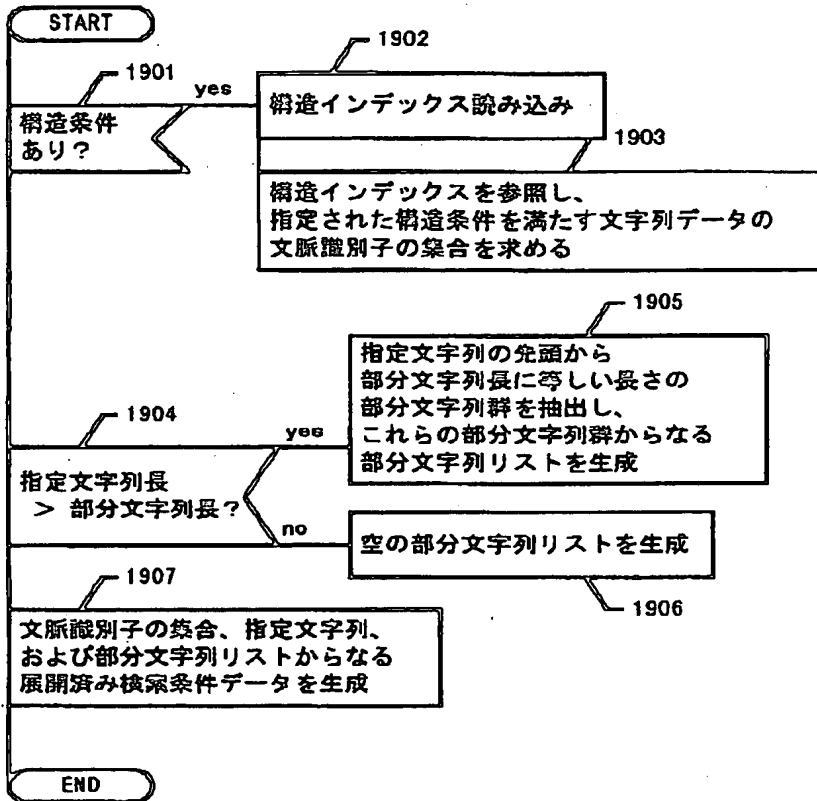
【図 18】

図 18

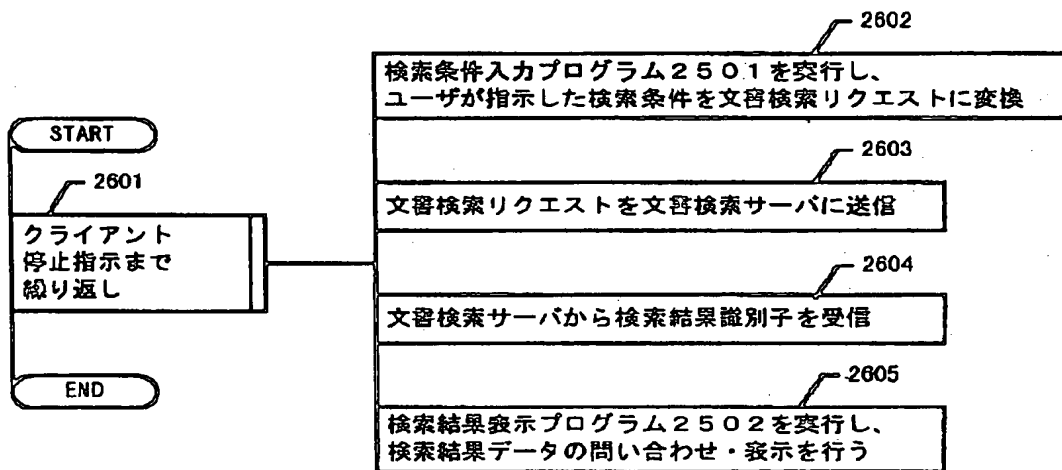


【図 1 9】

図 1 9

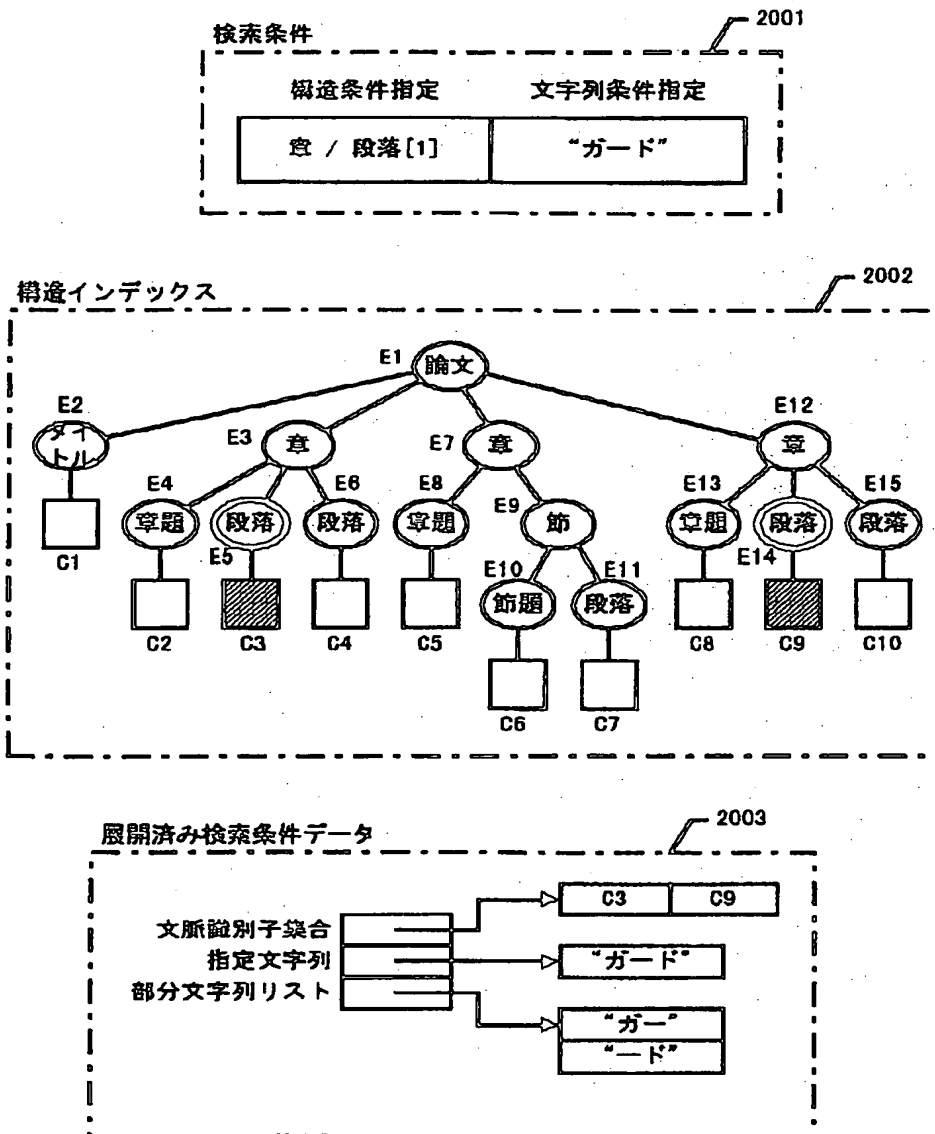


【図 2 6】

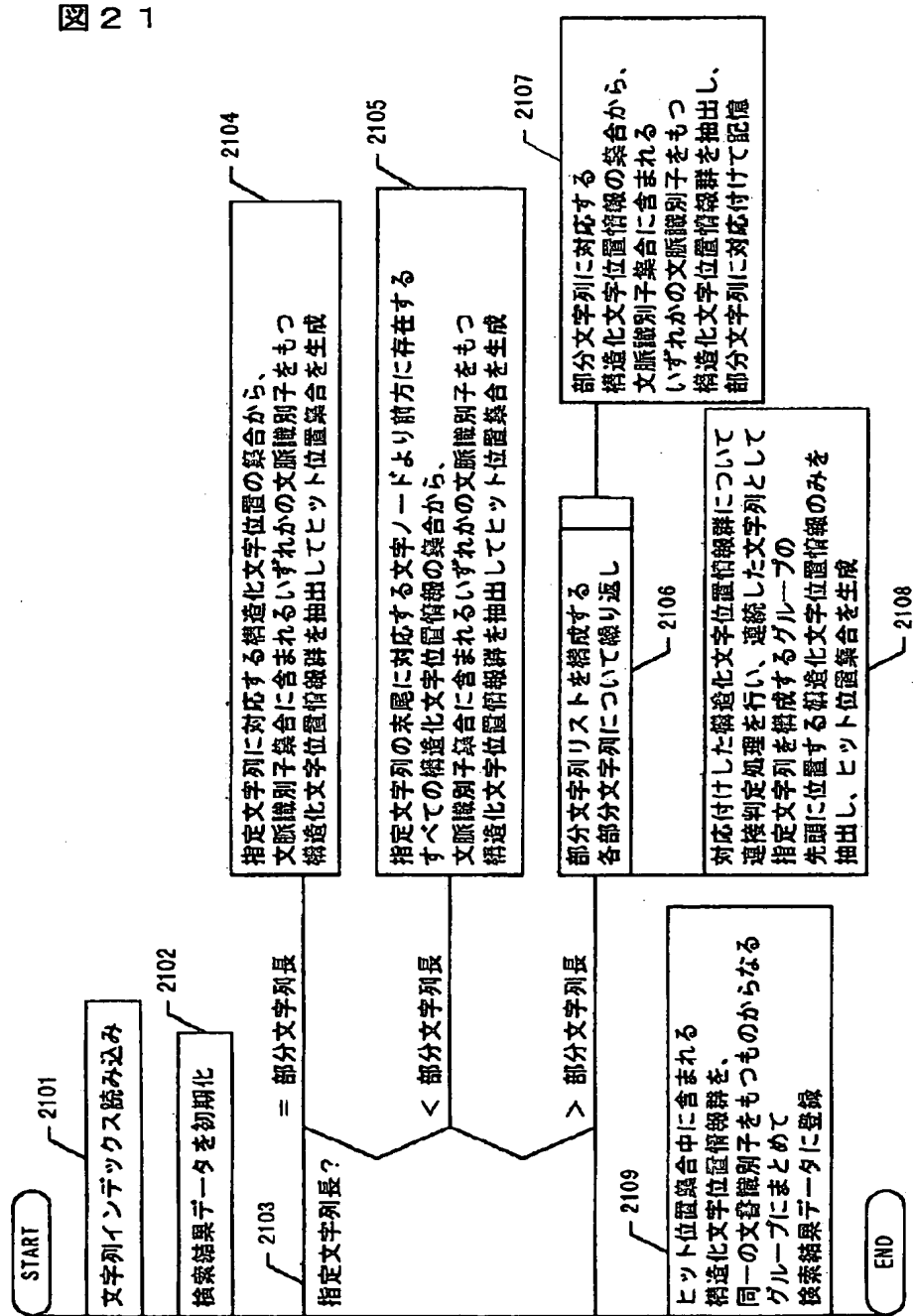


【図 20】

図 20

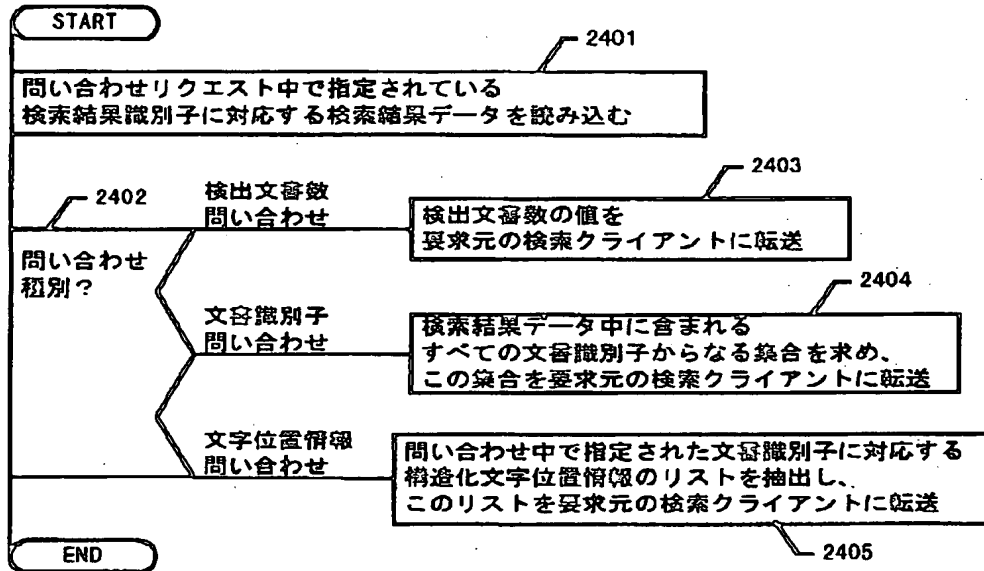


【図 2 1】

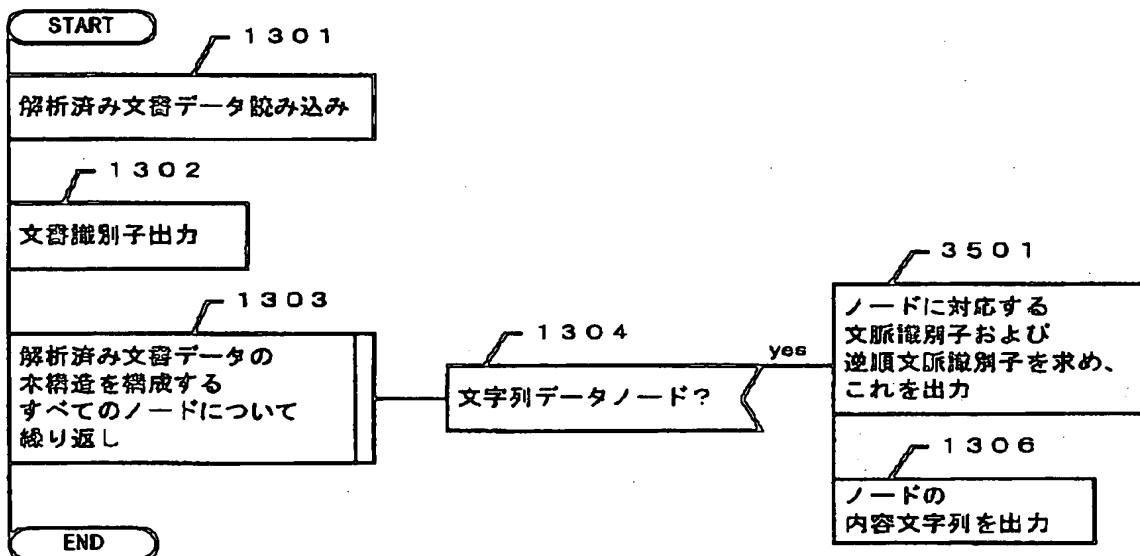


【図 24】

図 24

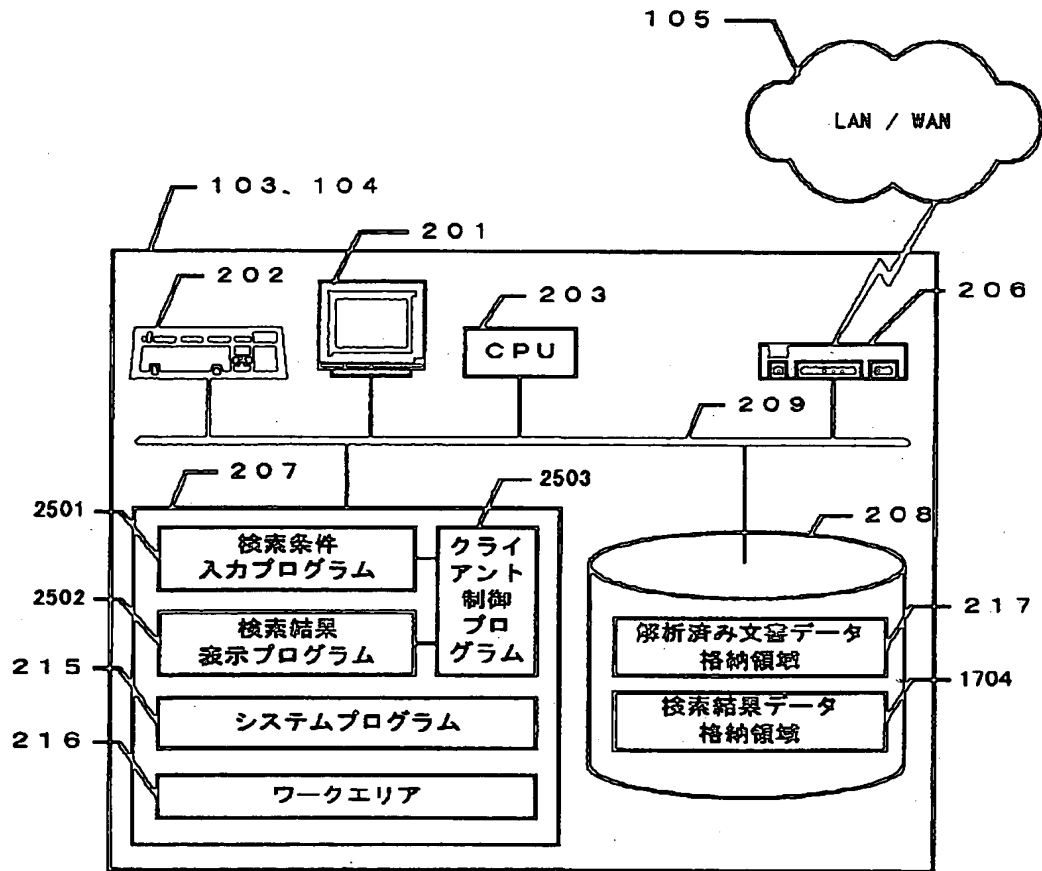


【図 35】



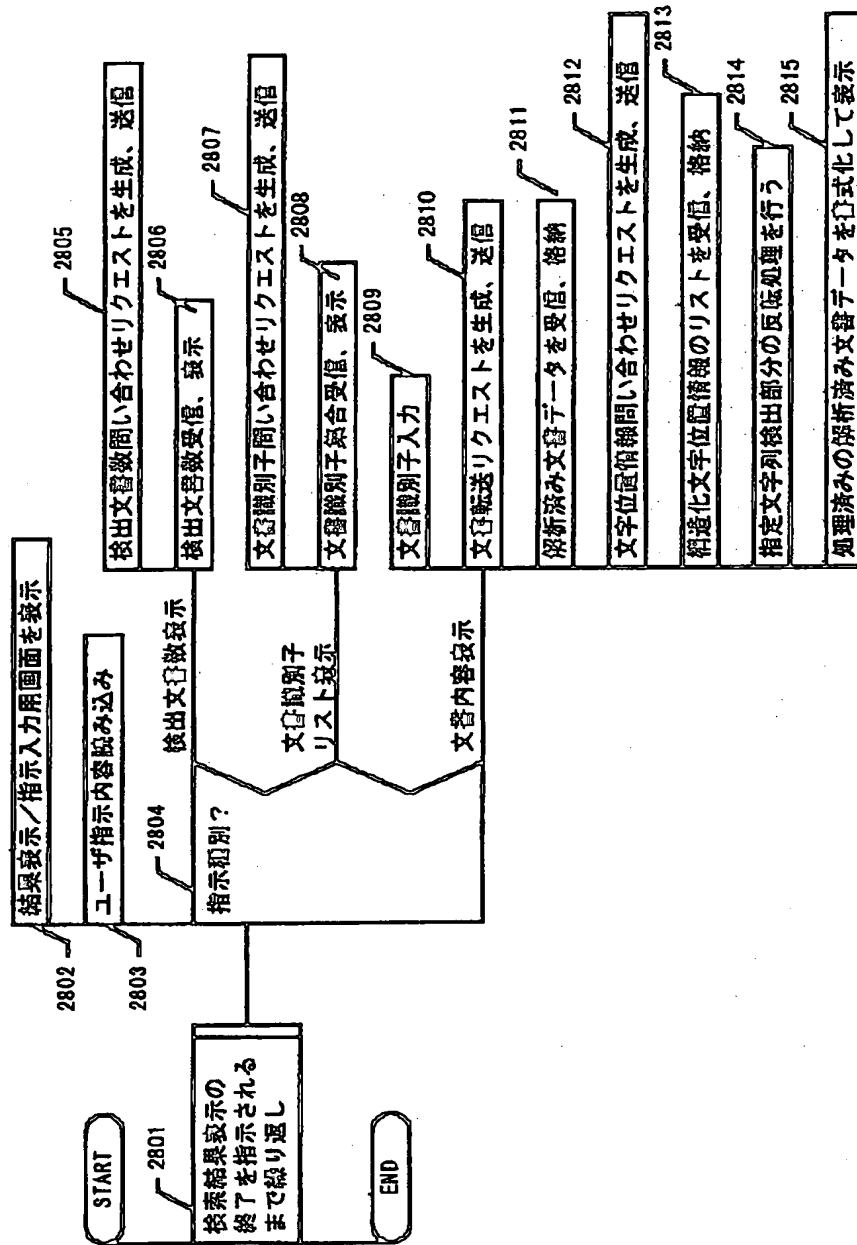
【図25】

図25



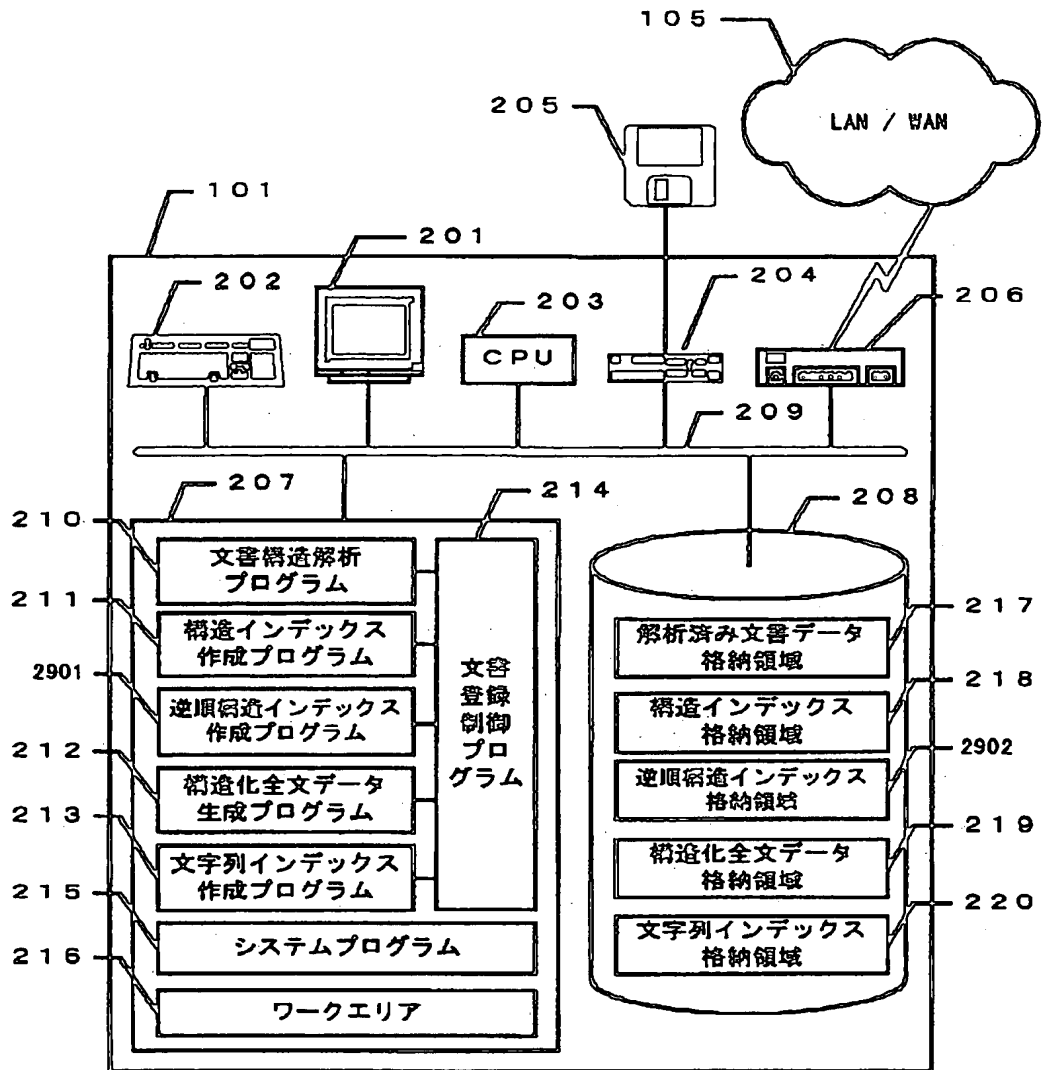
【図 2 8】

図 2 8



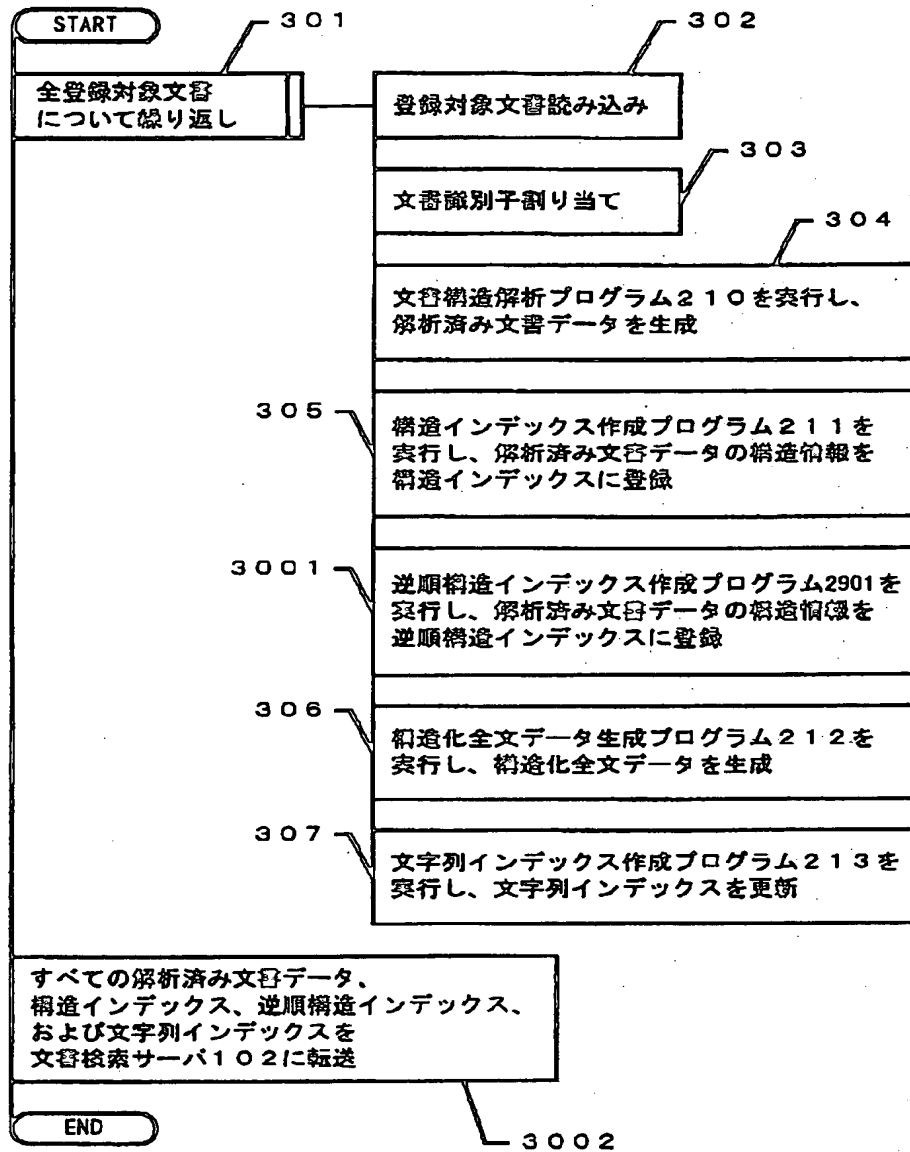
【図 29】

図 29



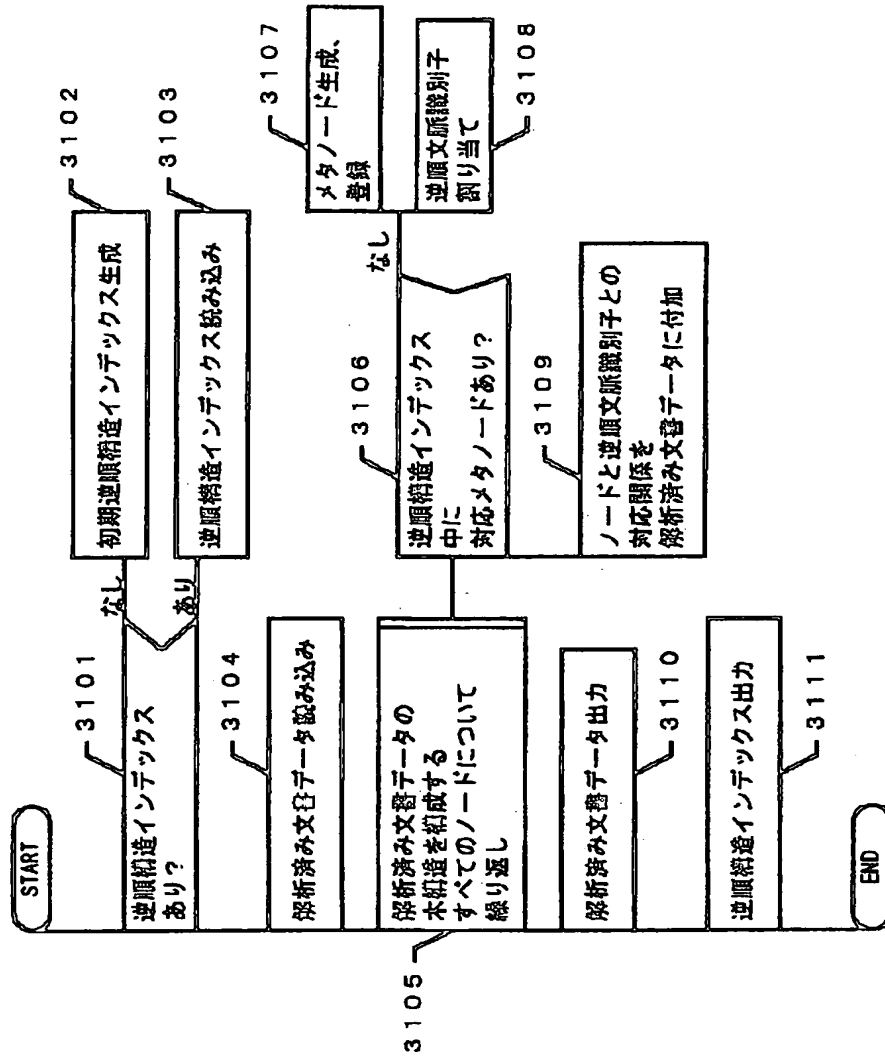
【図30】

図30



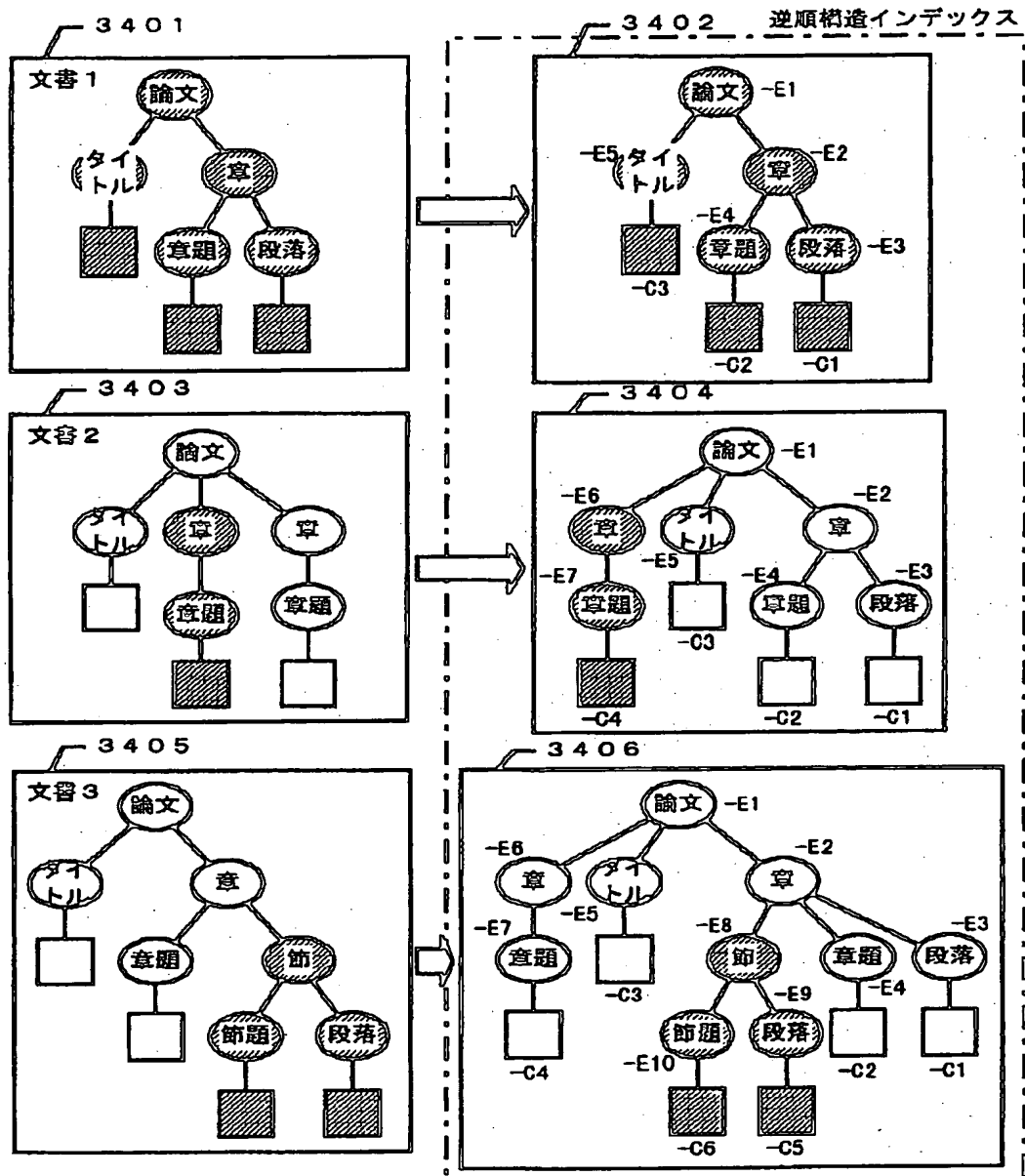
【図 3 1】

図 3 1



【図 34】

図 34



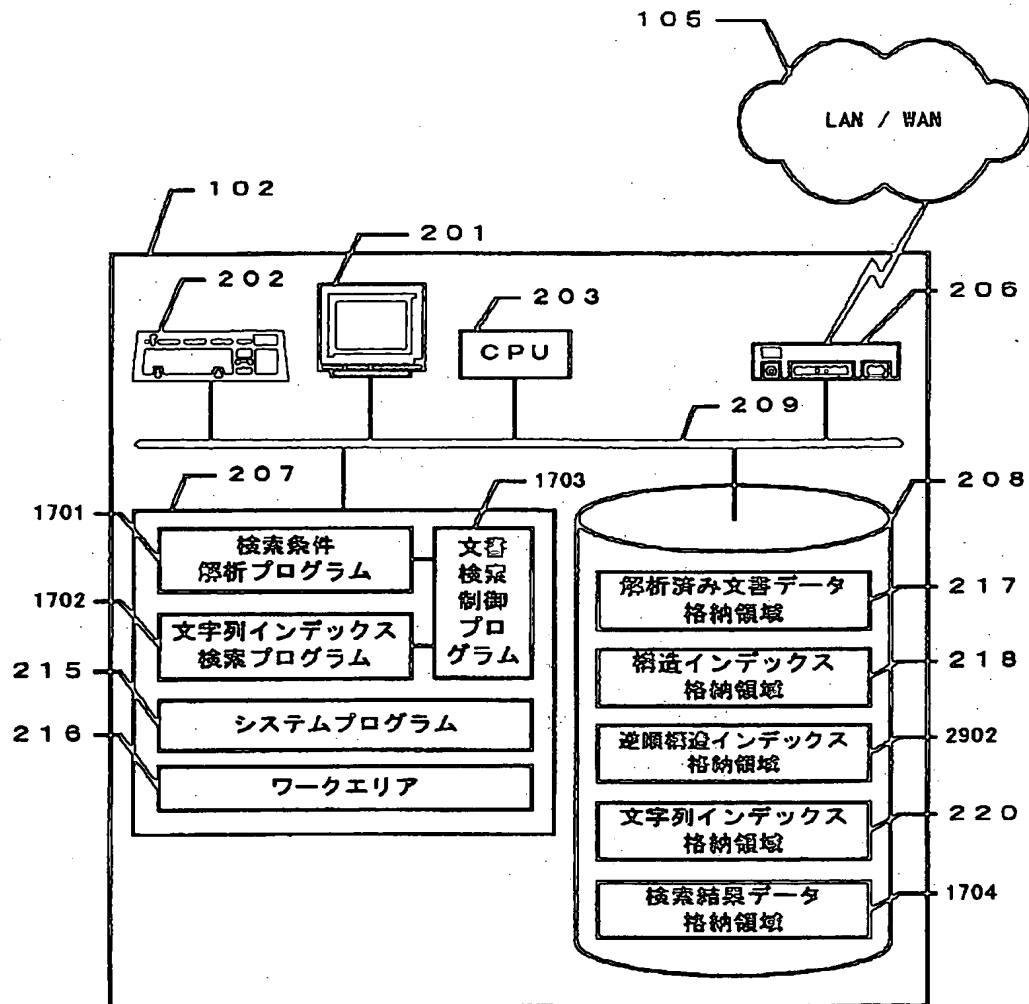
【図 3 6】

図 3 6

文番 識別子	文脈 識別子	逆順 文脈 識別子	内容文字列
D1	C1	-C351	“SGML文書変換言語の開発とその適用事例”
	C2	-C350	“高橋亨”
	C3	-C349	“東野純一”
	C4	-C348	“星串雄”
	C5	-C347	“1996年10月23日”
	C6	-C346	“はじめに”
	C7	-C345	“文書記述にSGMLを用いることによって…”
	C8	-C344	“作成したSGML文書をさまざまな…”
	C15	-C337	“適用事例”
	C16	-C336	“背景”
	C17	-C335	“現在、ISOでは…”
	C129	-C223	“変換処理の事例”
	C130	-C222	“数式の変換”
	C131	-C221	“JIS規格DTDでは、基本的には数式を…”
	C132	-C220	“ただし、行列式の場合には…”

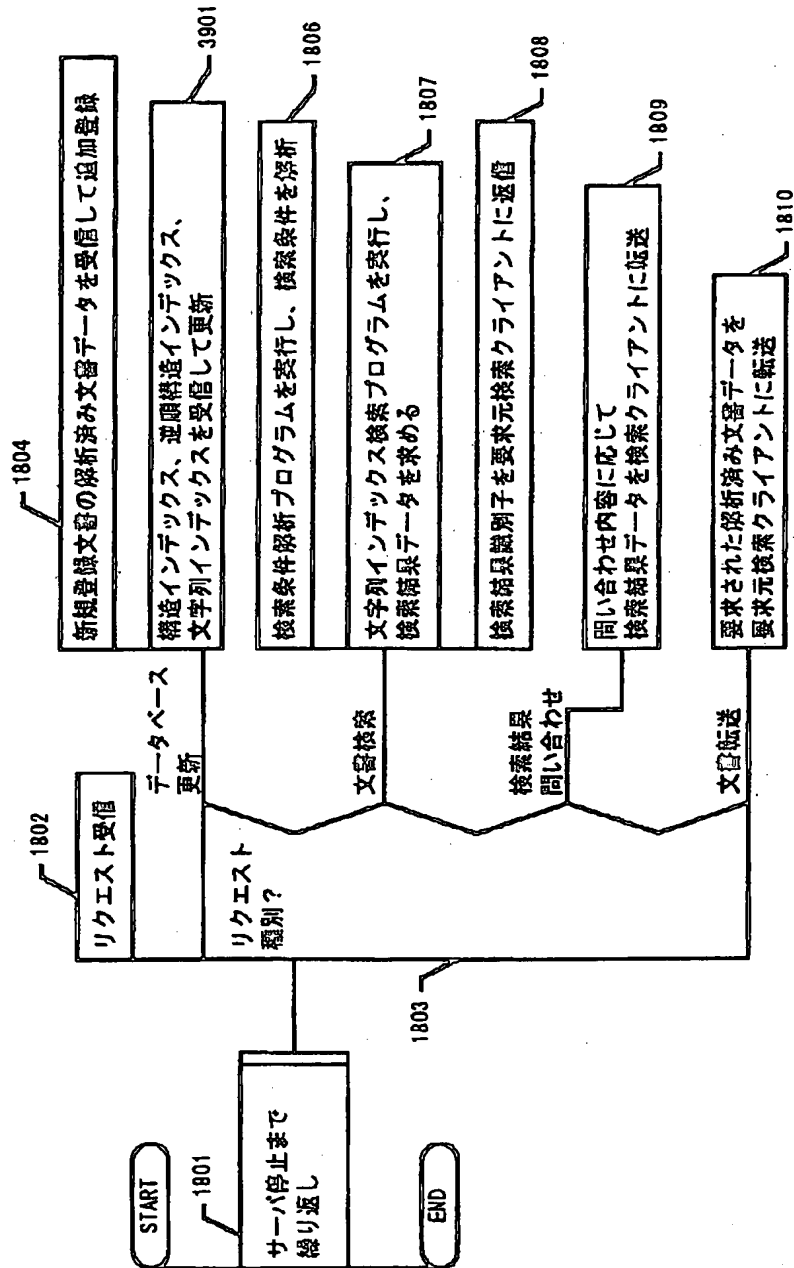
【図38】

図38



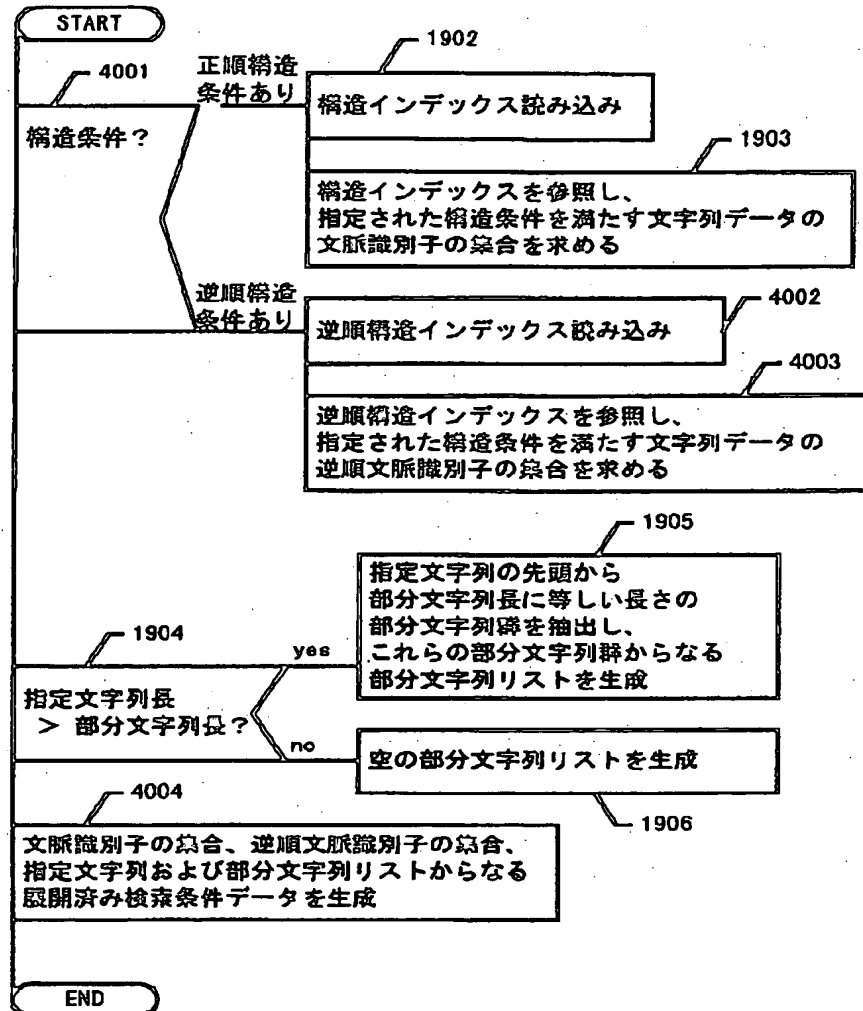
【図39】

図 39



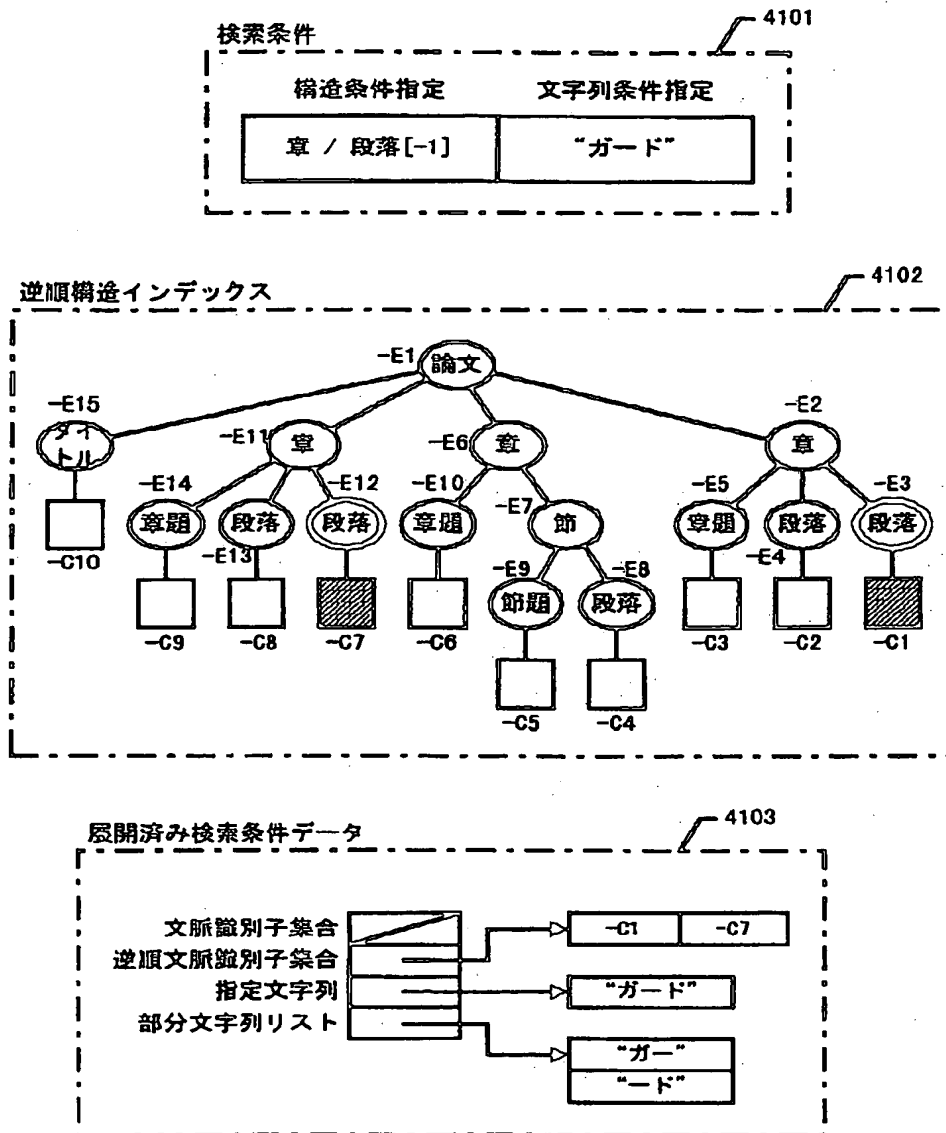
【図 40】

図 40



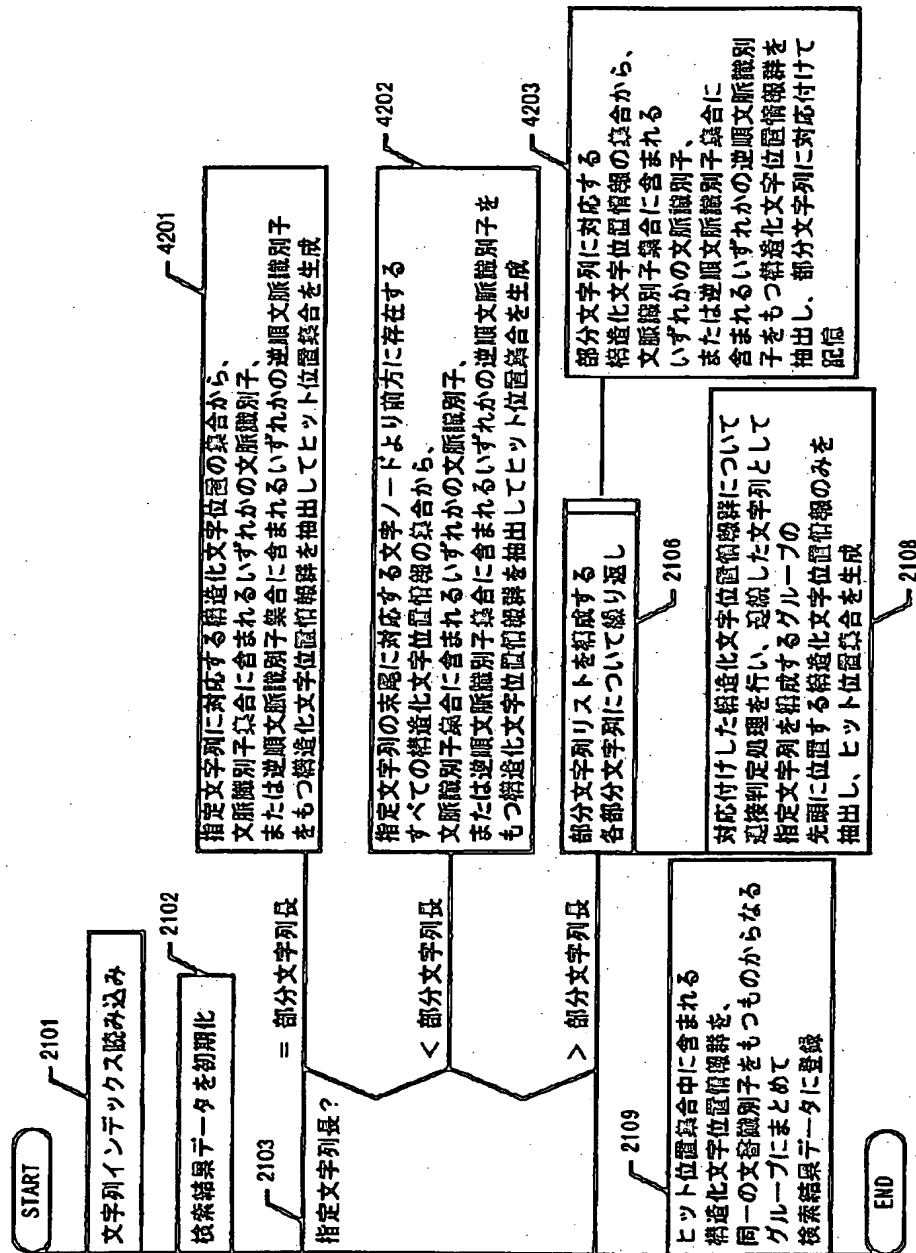
【図 4 1】

図 4 1



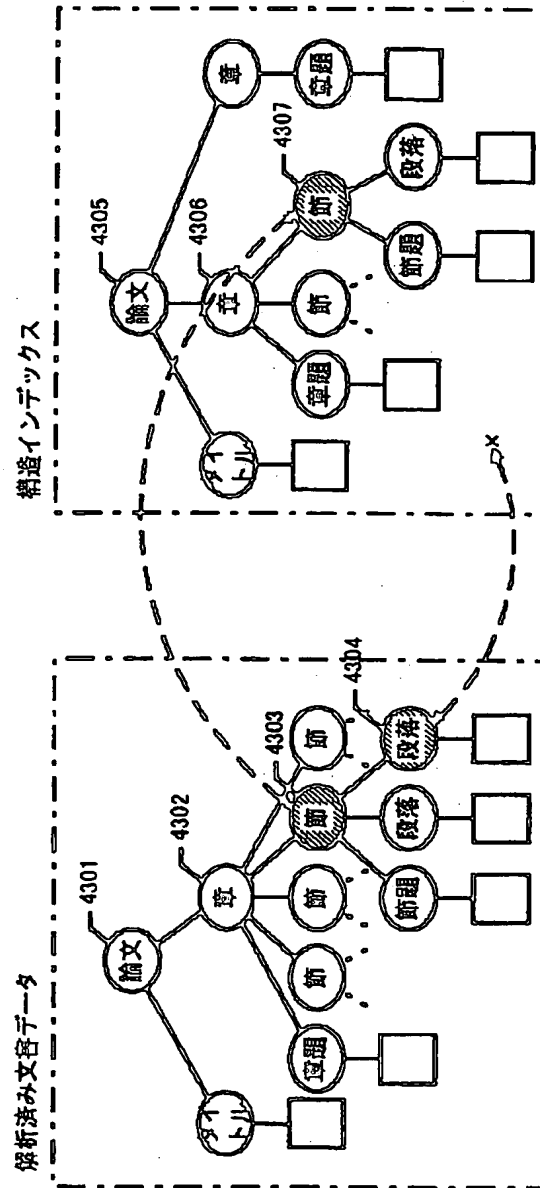
【図 4 2】

図 4 2



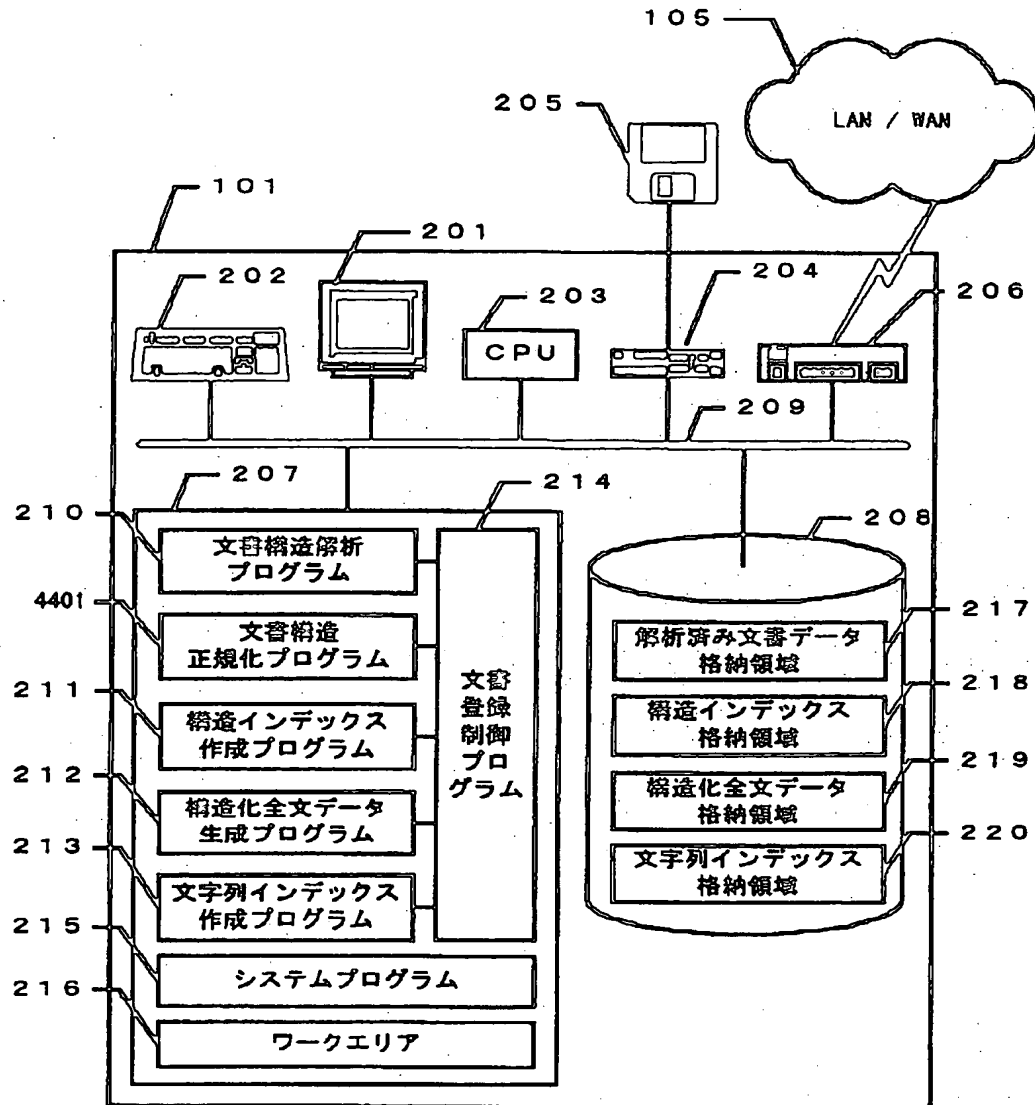
【図 4 3】

図 4 3



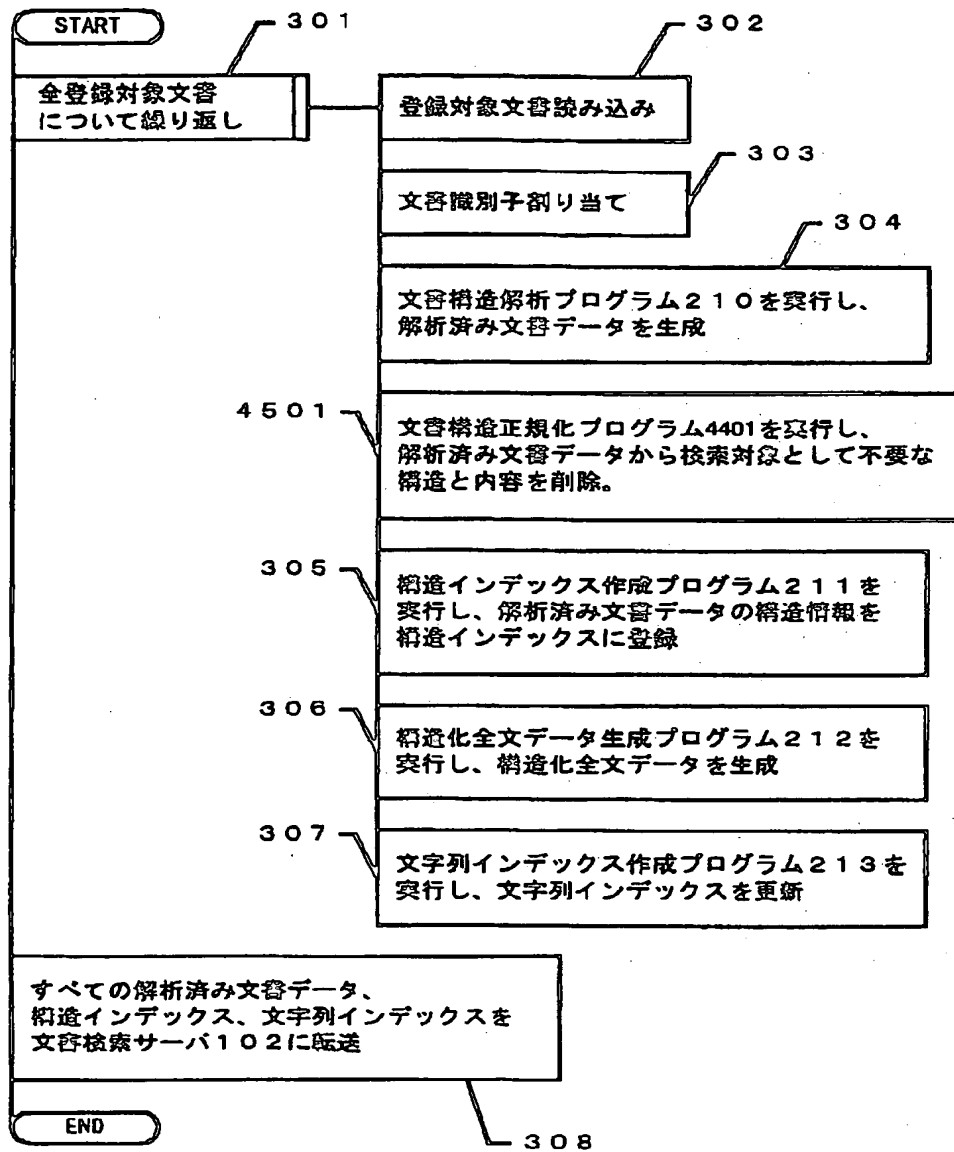
【図44】

図44



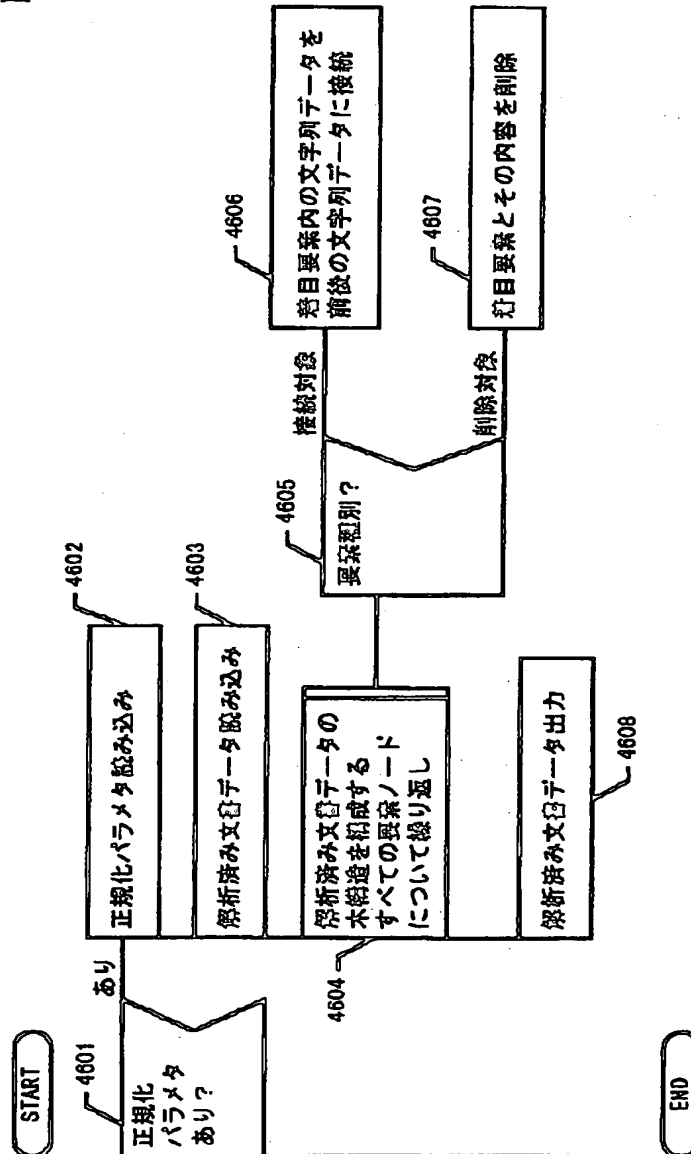
【図45】

図45



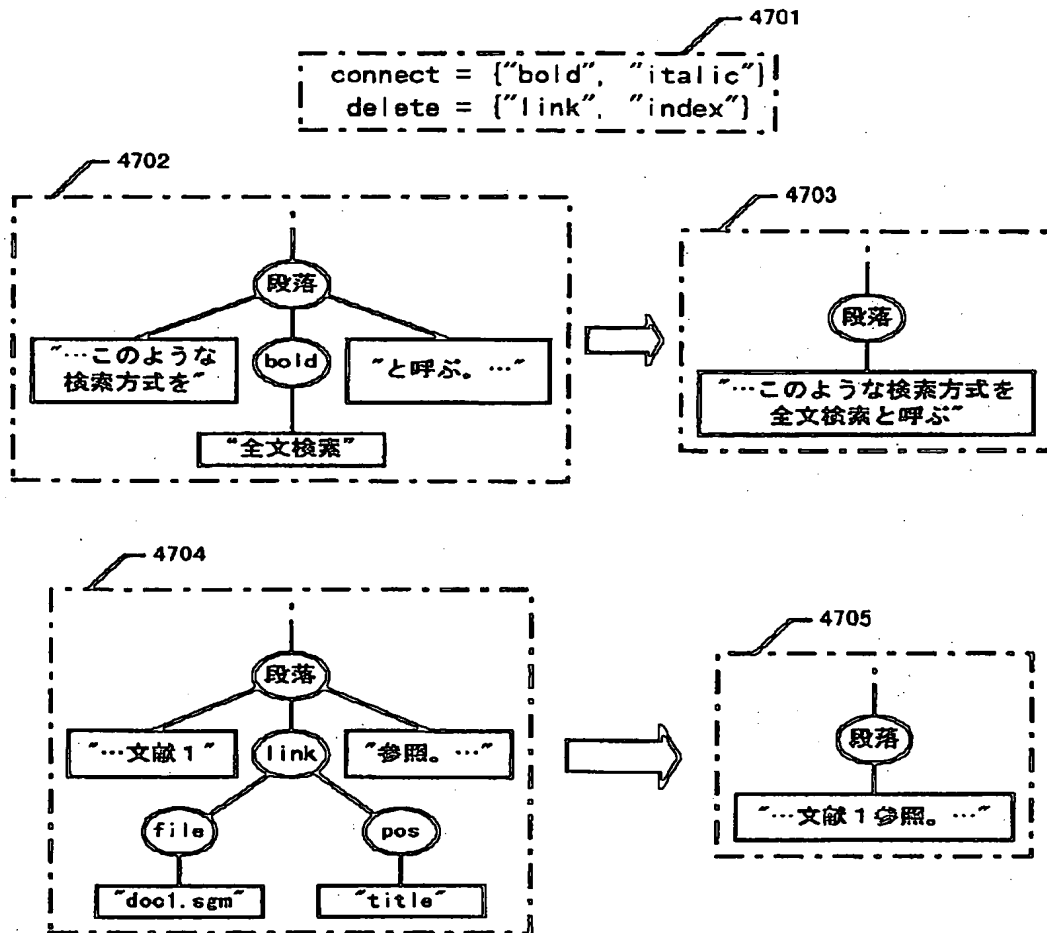
【図46】

図46



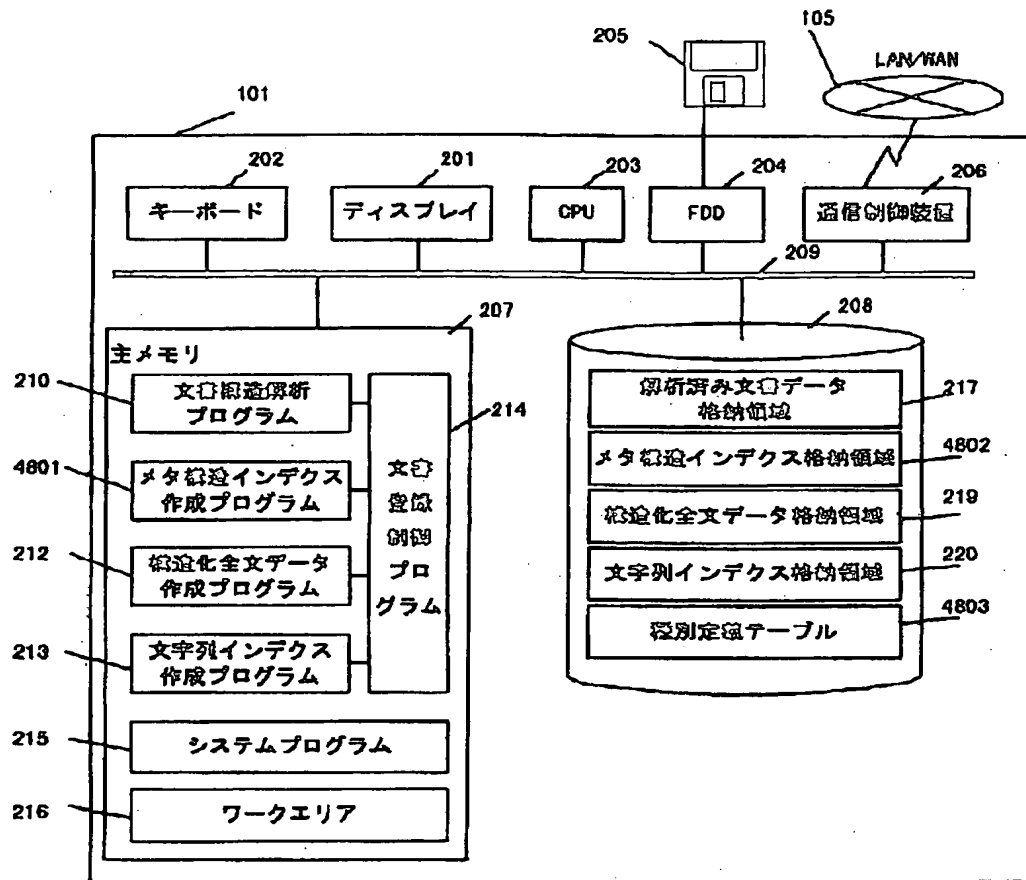
【図 4 7】

図 4 7

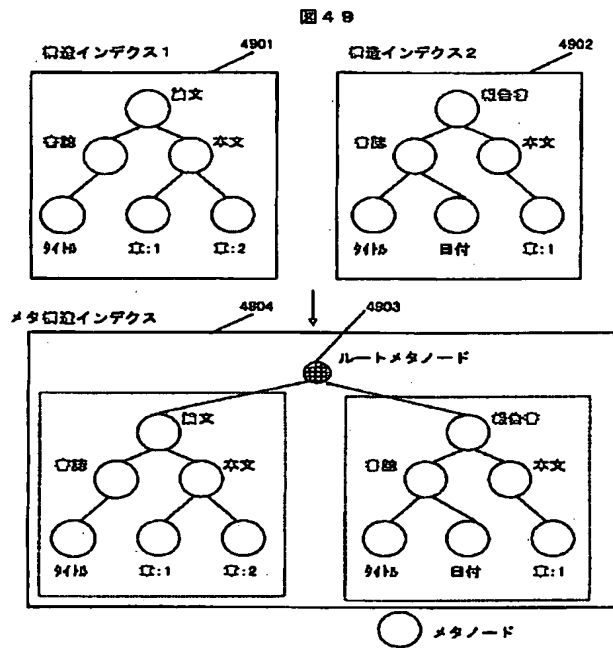


【図48】

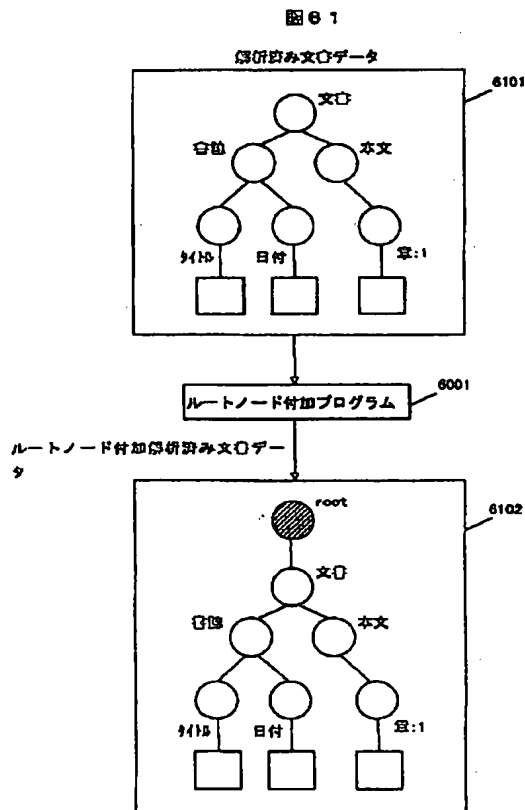
図48



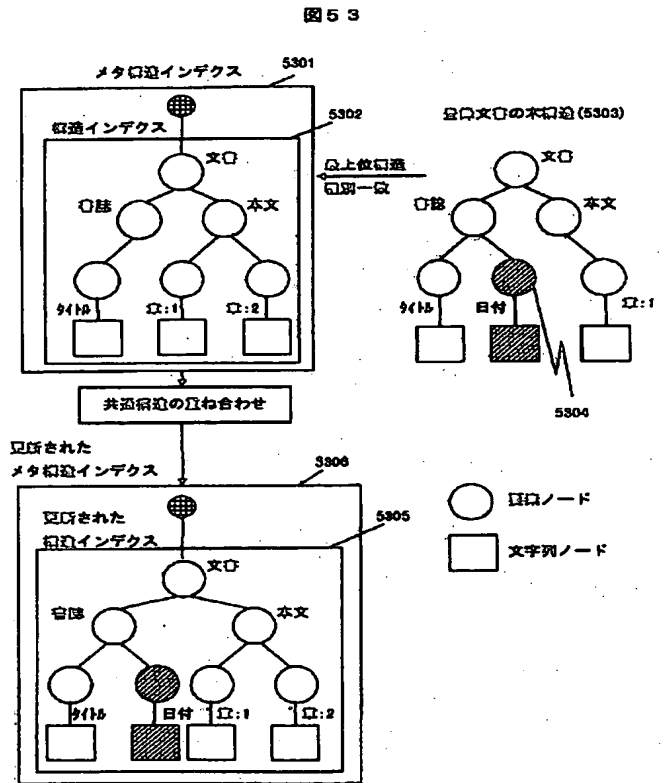
【図49】



【図61】

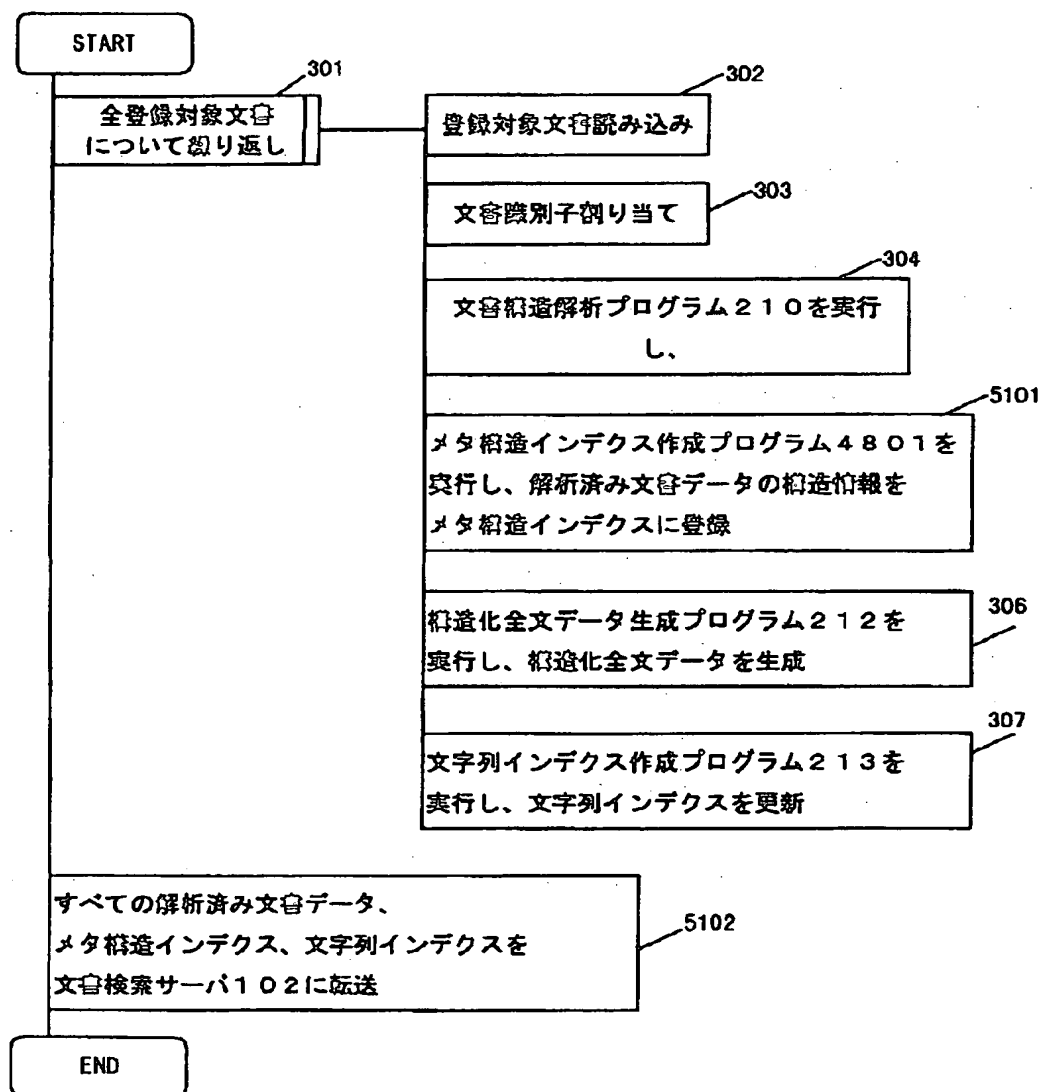


【図53】



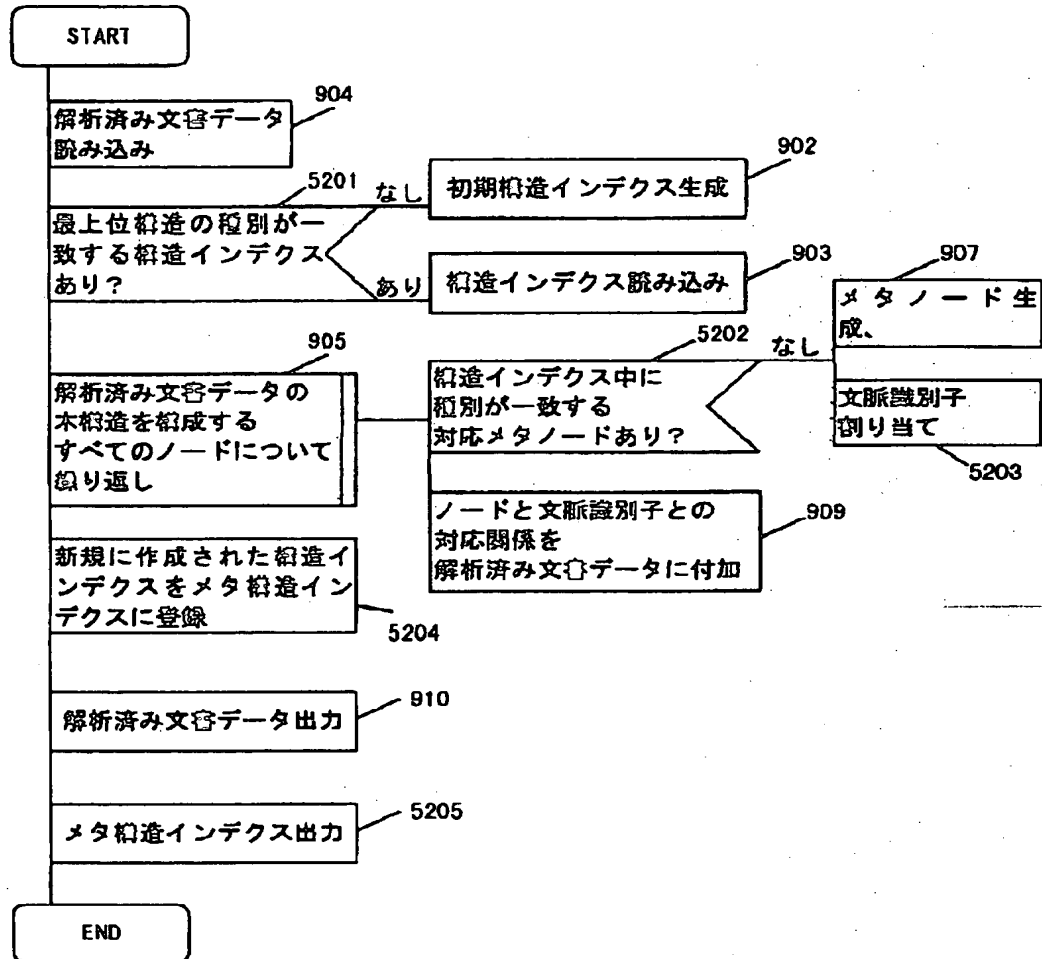
【図51】

図51



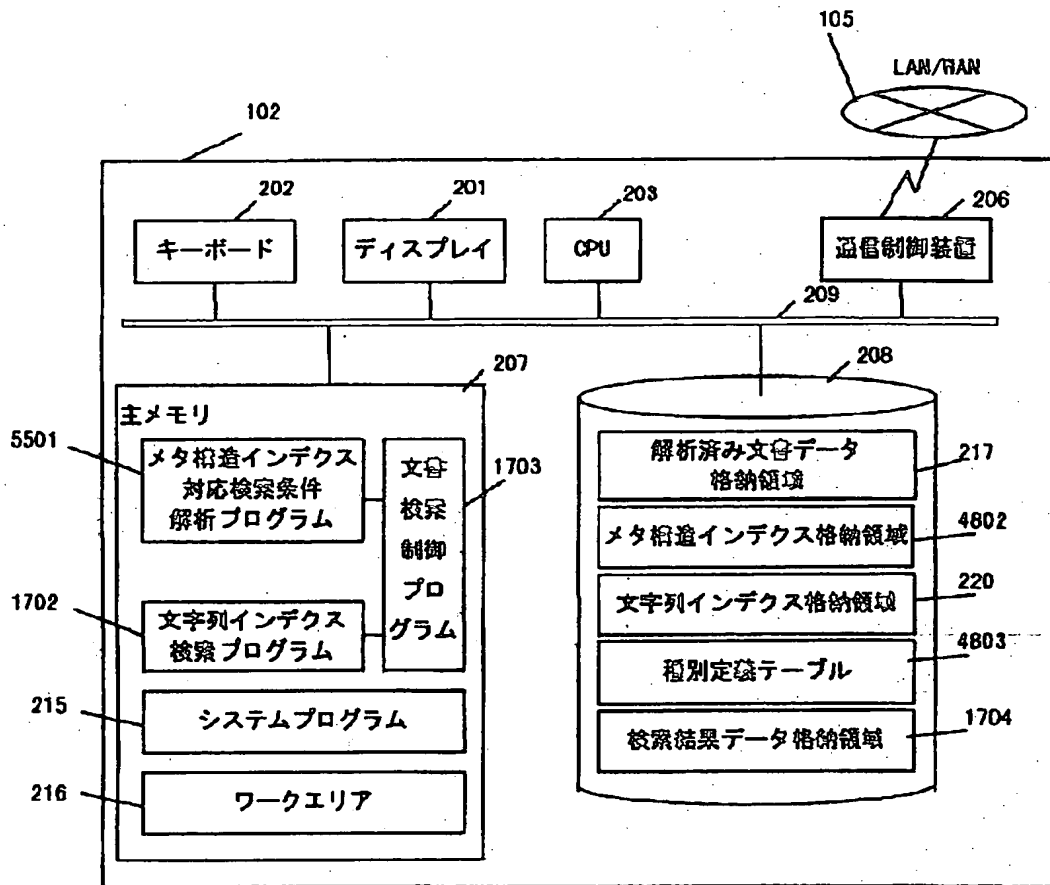
【図52】

図52



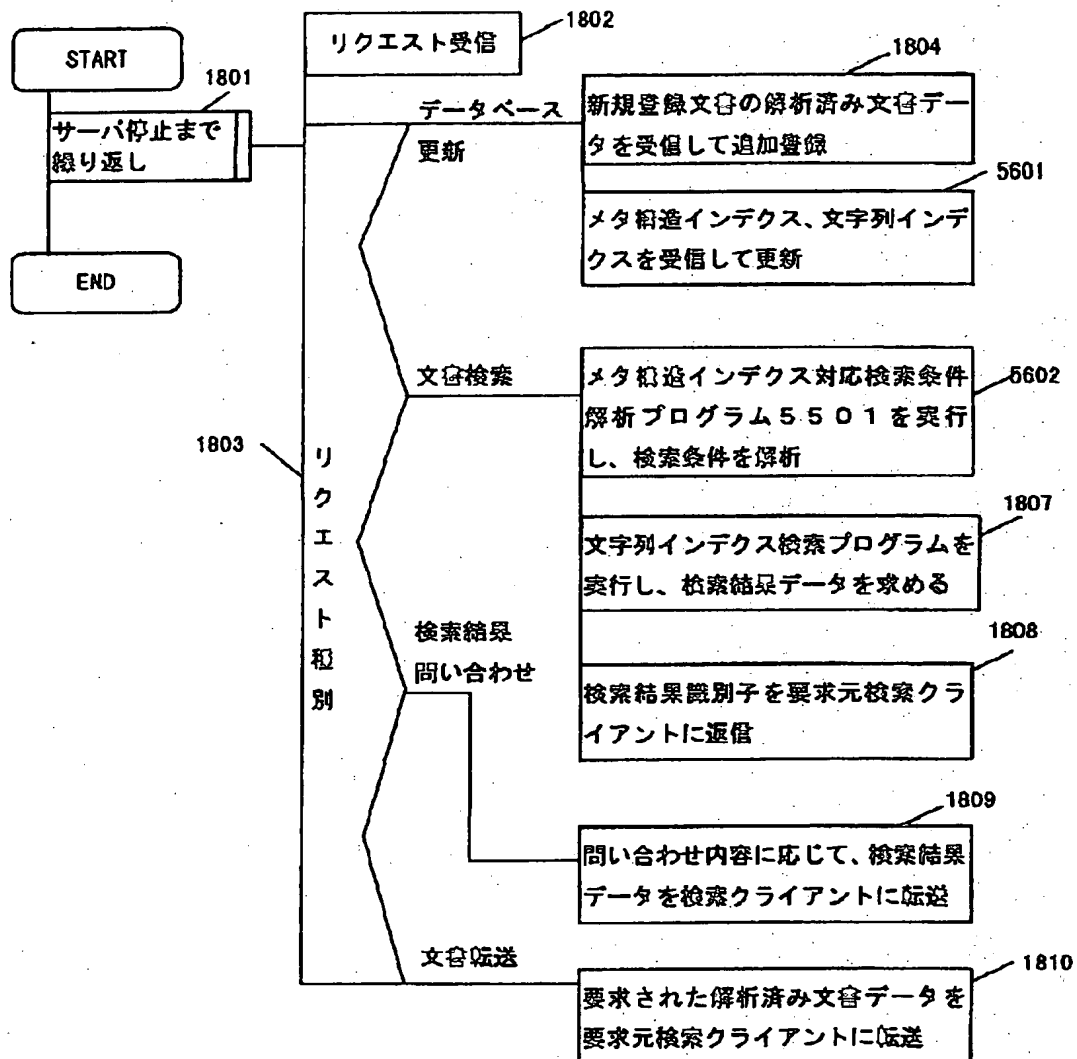
【図55】

図55



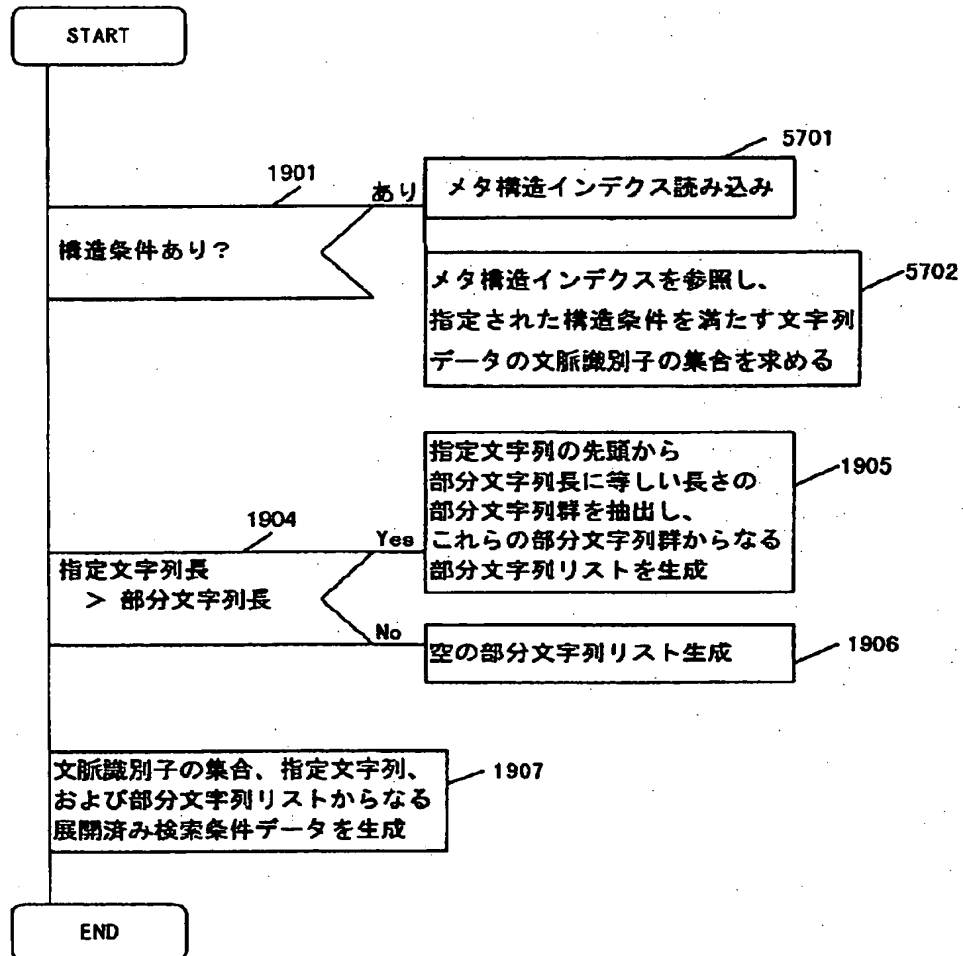
【図 5 6】

図 5 6



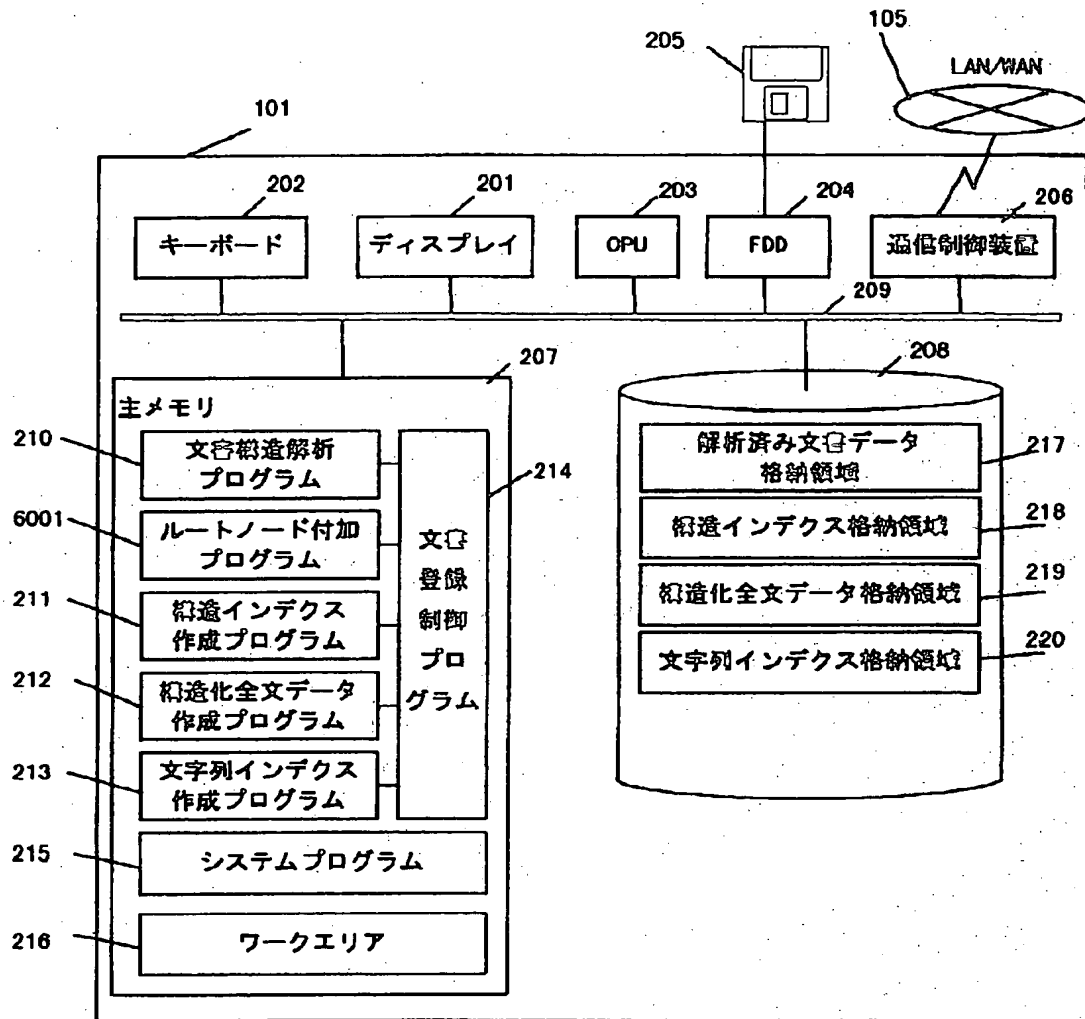
【図 5 7】

図 5 7



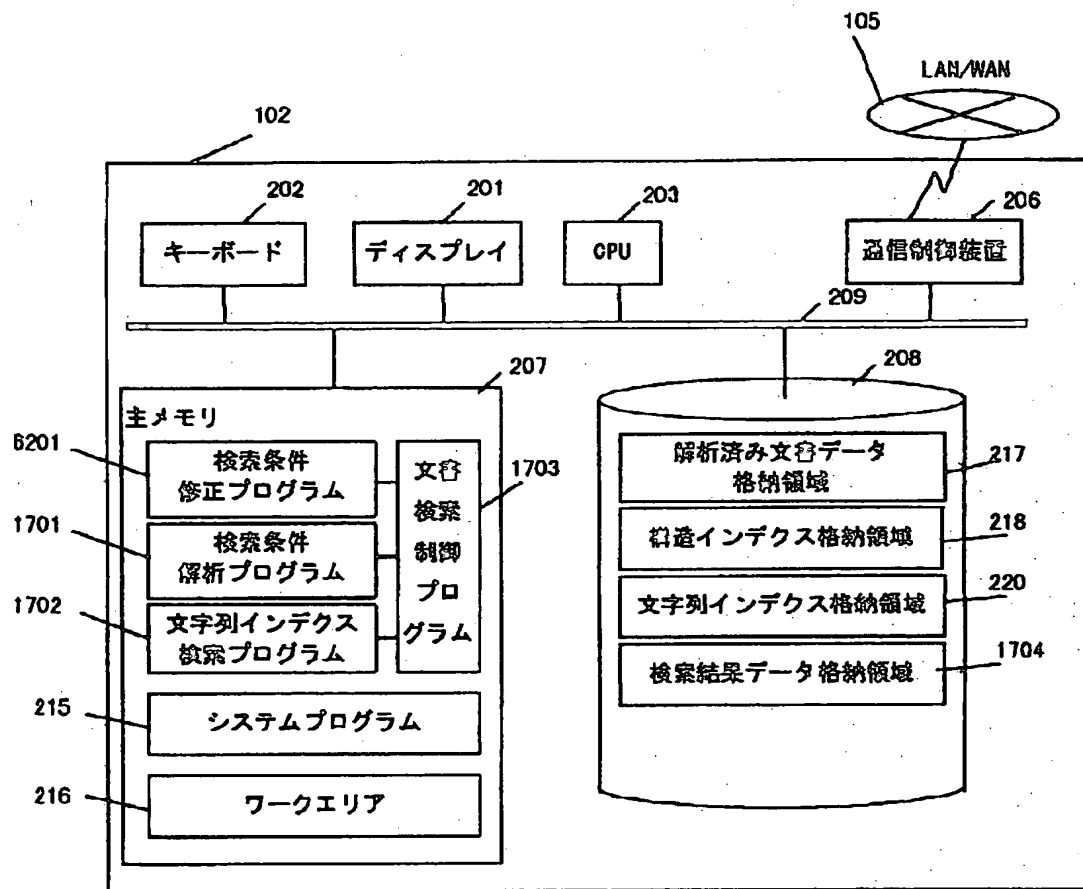
【図 60】

図 60



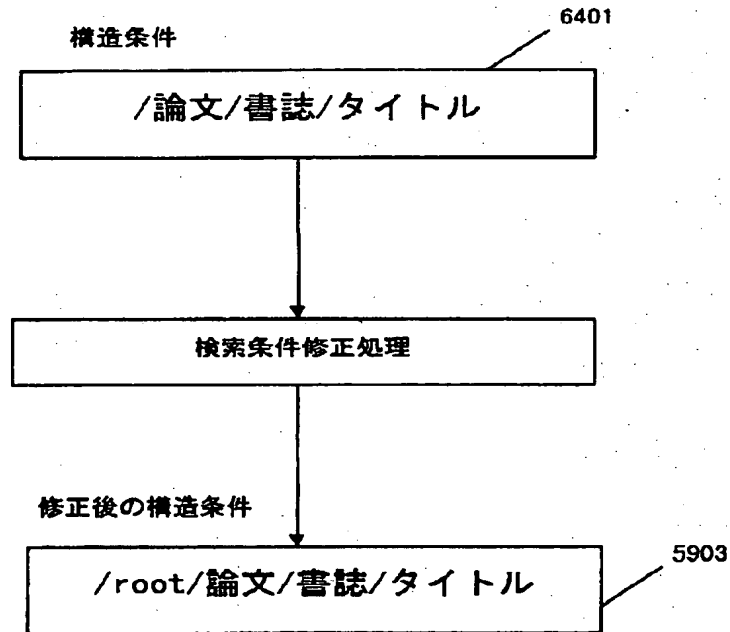
【図 6 2】

図 6 2



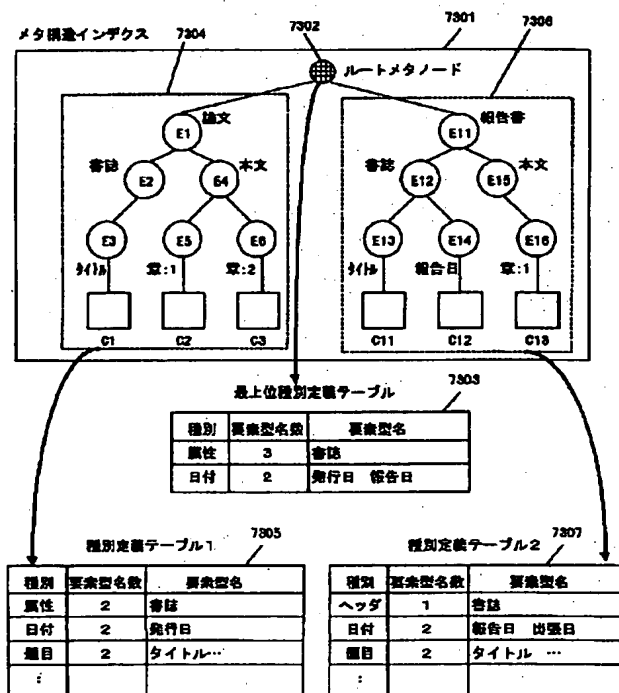
【図64】

図64

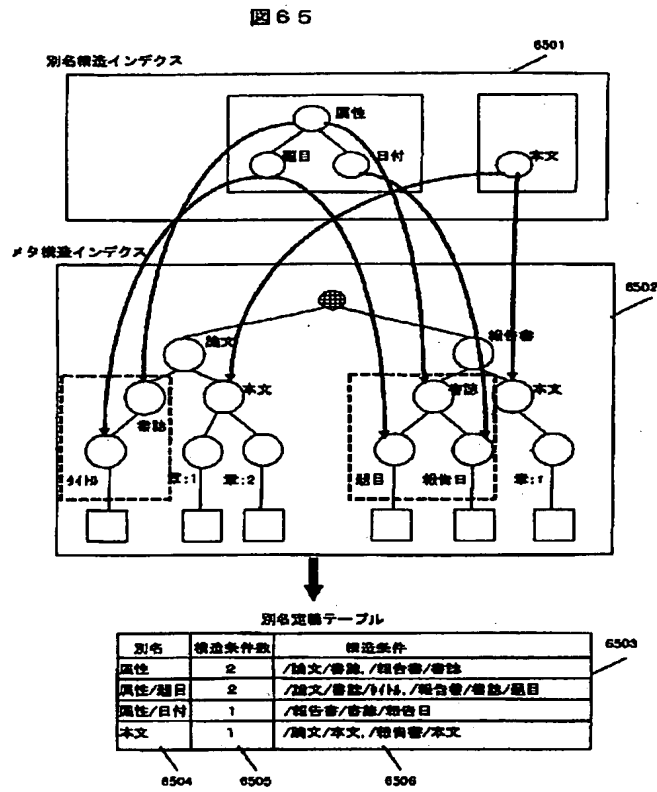


【図73】

図73

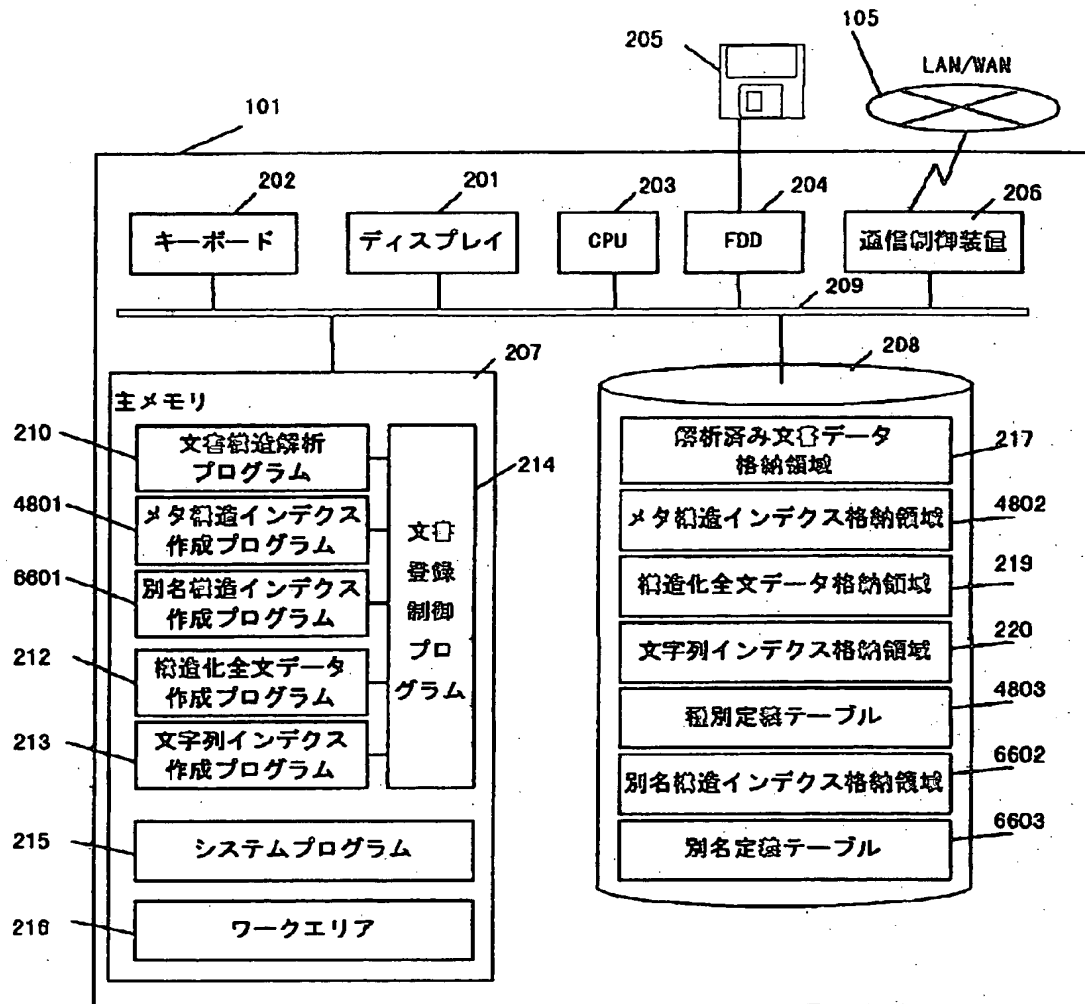


【図 65】



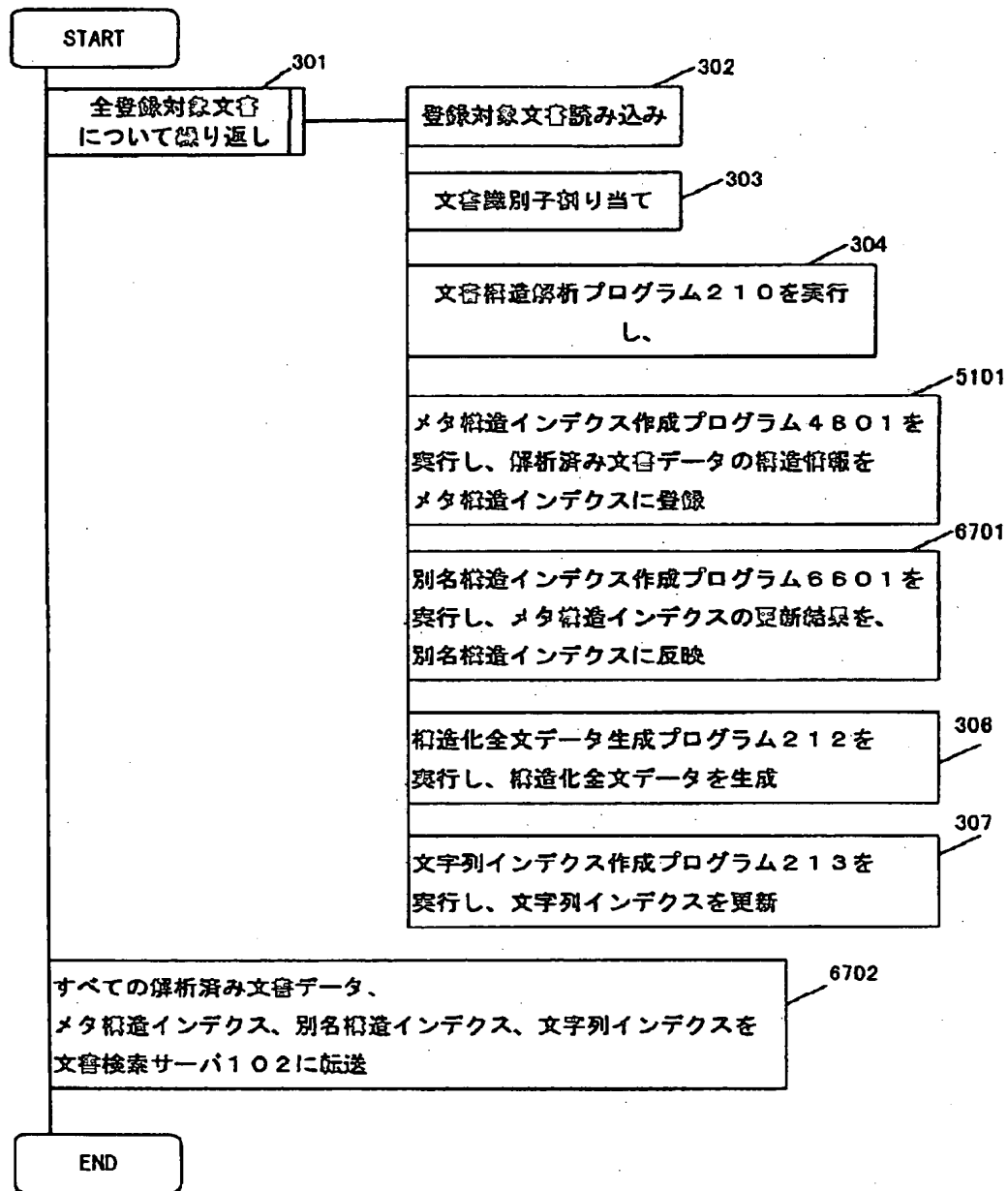
【図 6 6】

図 6 6



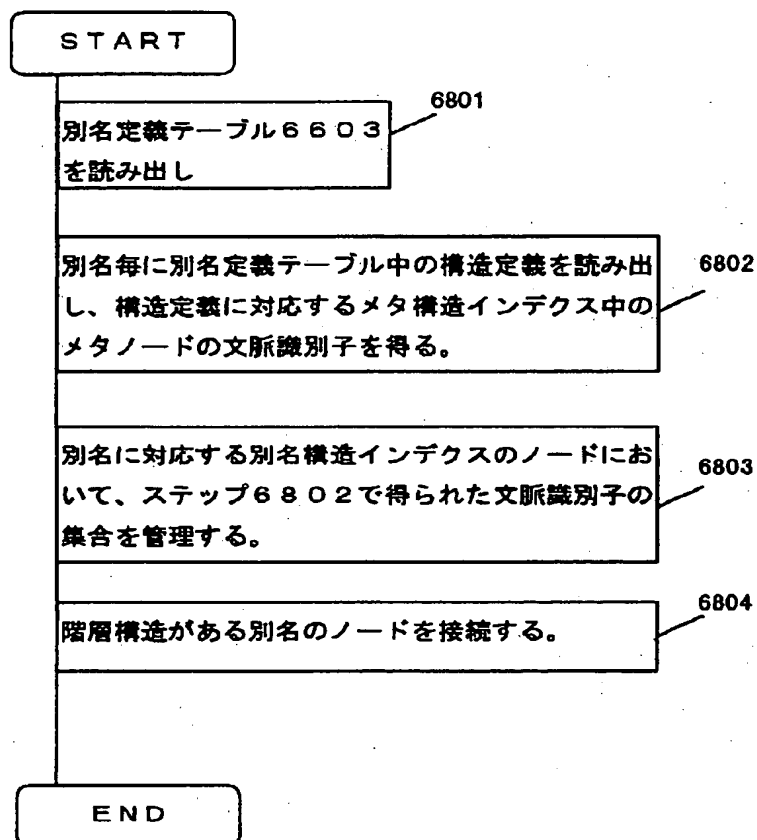
【図 6 7】

図 6 7



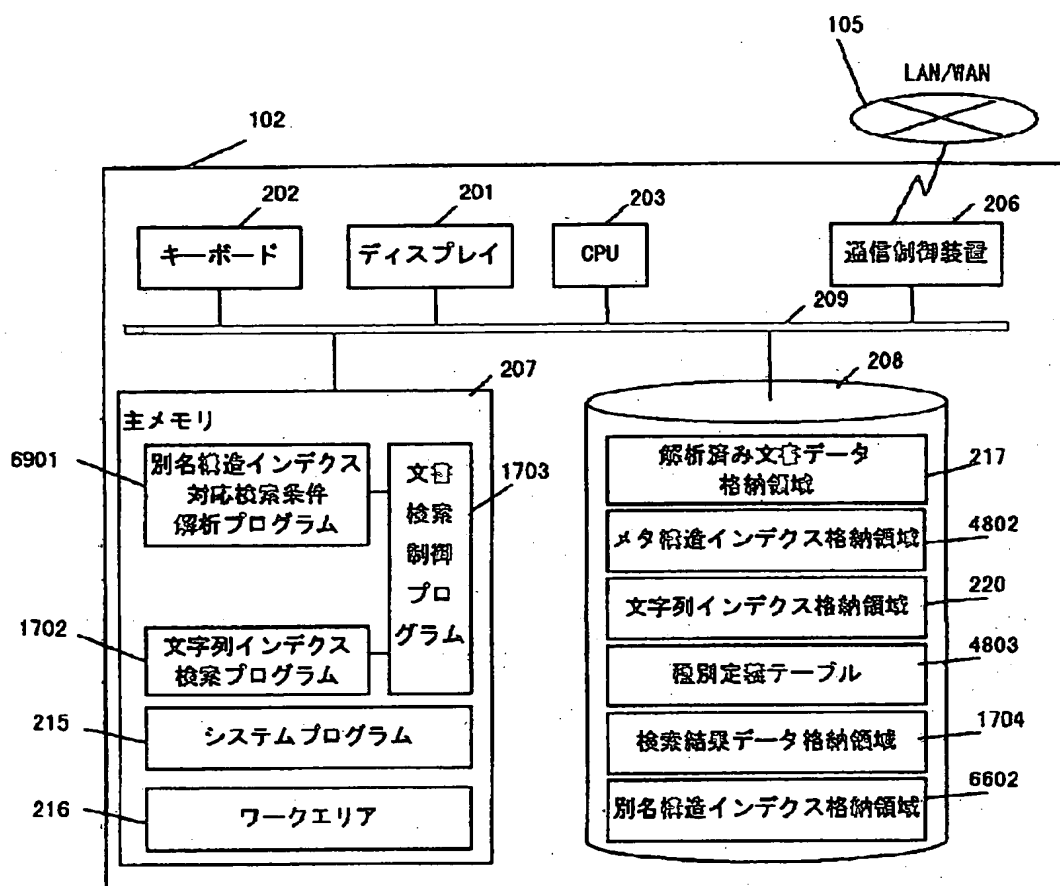
【図 6 8】

図 6 8



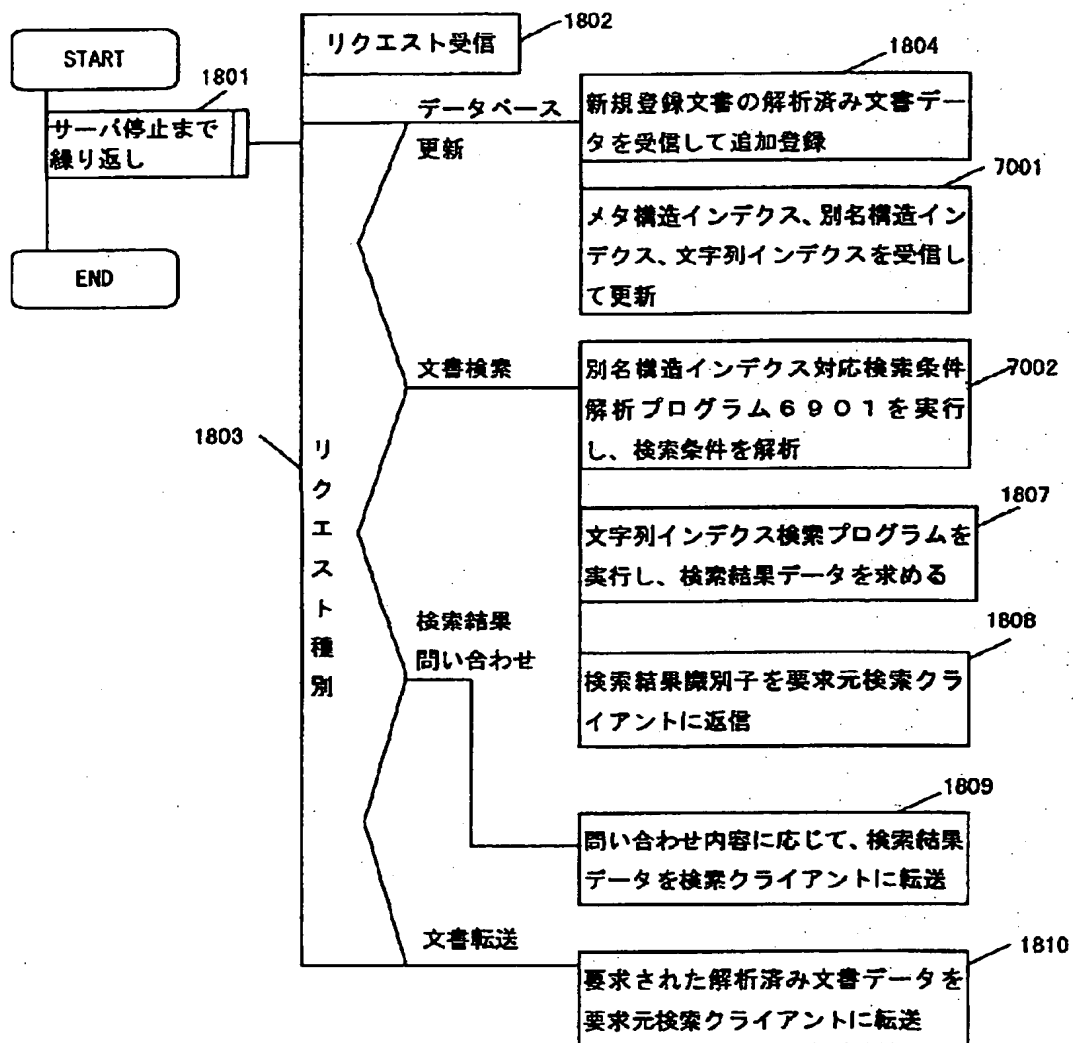
【図 6 9】

図 6 9



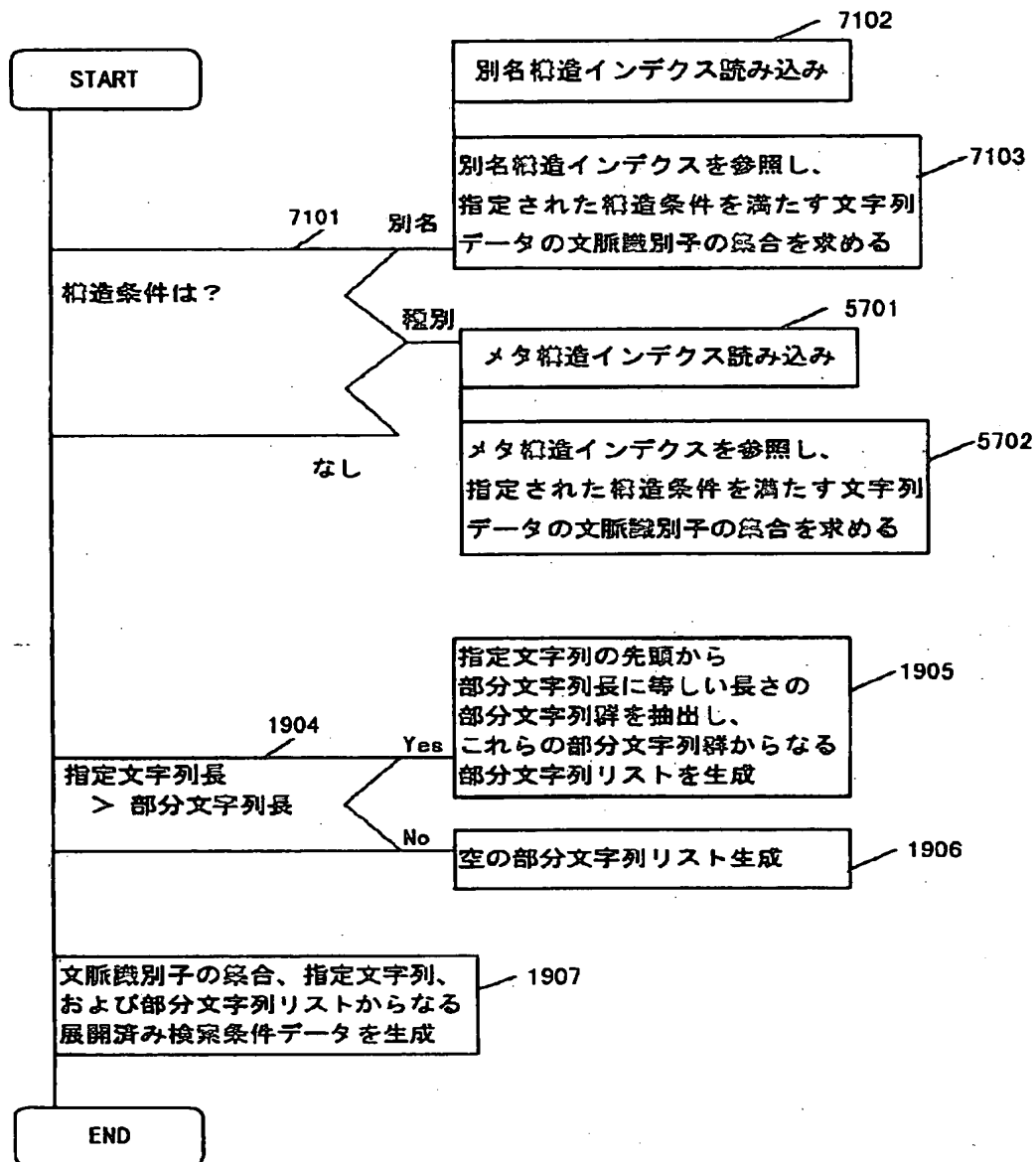
【図 70】

図 70



【図71】

図71



フロントページの続き

(72)発明者 多田 勝己
 神奈川県川崎市幸区鹿島田890番地 株式
 会社日立製作所情報通信開発本部内

(72)発明者 山崎 紀之
 神奈川県横浜市戸塚区戸塚町5030番地 株
 式会社日立製作所ソフトウェア開発本部内